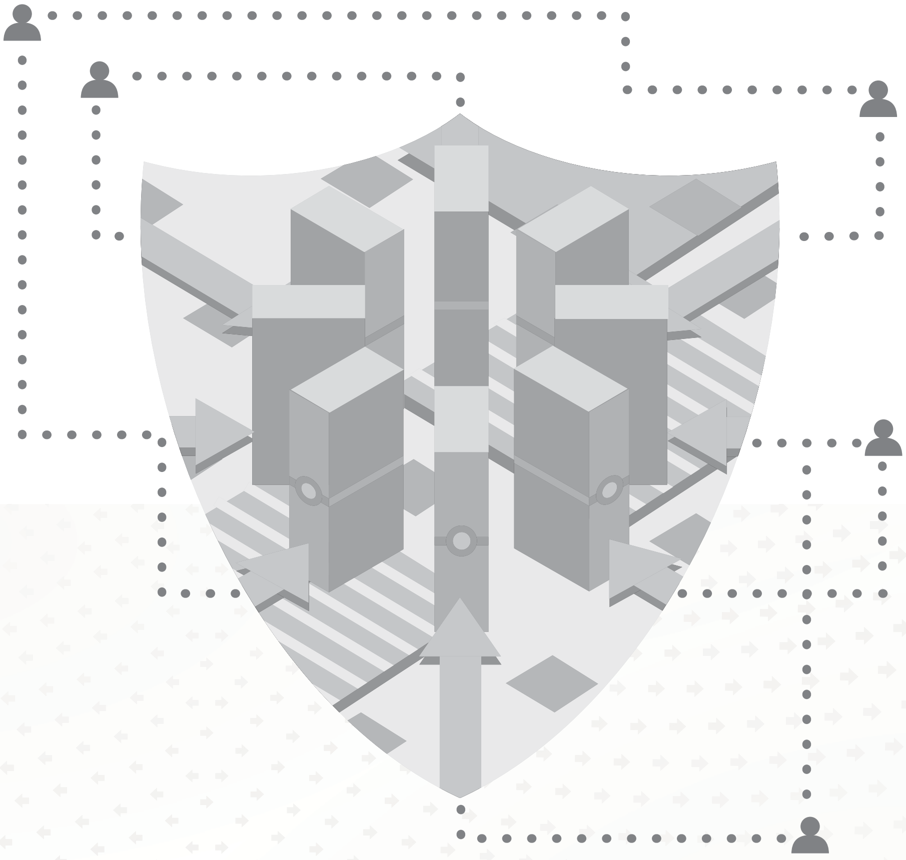
Sourcefire 3D System

eStreamer Integration Guide

Version 5.3



Sourcefire is now part of Cisco.

Legal Notices

Cisco, the Cisco logo, Sourcefire, the Sourcefire logo, Snort, the Snort and Pig logo, and certain other trademarks and logos are trademarks or registered trademarks of Cisco and/or its affiliates in the United States and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

The legal notices, disclaimers, terms of use, and other information contained herein (the "terms") apply only to the information discussed in this documentation (the "Documentation") and your use of it. These terms do not apply to or govern the use of websites controlled by Cisco or its subsidiaries (collectively, "Cisco") or any Sourcefire-provided or Cisco-provided products. Sourcefire and Cisco products are available for purchase and subject to a separate license agreement and/or terms of use containing very different terms and conditions.

The copyright in the Documentation is owned by Cisco and is protected by copyright and other intellectual property laws of the United States and other countries. You may use, print out, save on a retrieval system, and otherwise copy and distribute the Documentation solely for non-commercial use, provided that you (i) do not modify the Documentation in any way and (ii) always include Cisco's copyright, trademark, and other proprietary notices, as well as a link to, or print out of, the full contents of this page and its terms.

No part of the Documentation may be used in a compilation or otherwise incorporated into another work or with or into any other documentation or user manuals, or be used to create derivative works, without the express prior written permission of Cisco. Cisco reserves the right to change the terms at any time, and your continued use of the Documentation shall be deemed an acceptance of those terms.

Disclaimers

THE DOCUMENTATION AND ANY INFORMATION AVAILABLE FROM IT MAY INCLUDE INACCURACIES OR TYPOGRAPHICAL ERRORS. CISCO MAY CHANGE THE DOCUMENTATION FROM TIME TO TIME. CISCO MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE ACCURACY OR SUITABILITY OF ANY CISCO-CONTROLLED WEBSITE, THE DOCUMENTATION AND/OR ANY PRODUCT INFORMATION. CISCO-CONTROLLED WEBSITES, THE DOCUMENTATION AND ALL PRODUCT INFORMATION ARE PROVIDED "AS IS" AND CISCO DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO WARRANTIES OF TITLE AND THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL CISCO BE LIABLE TO YOU FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE, OR CONSEQUENTIAL DAMAGES (INCLUDING BUT NOT LIMITED TO PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF DATA, LOSS OF PROFITS, AND/OR BUSINESS INTERRUPTIONS), ARISING OUT OF OR IN ANY WAY RELATED TO CISCO-CONTROLLED WEBSITES OR THE DOCUMENTATION, NO MATTER HOW CAUSED AND/OR WHETHER BASED ON CONTRACT, STRICT LIABILITY, NEGLIGENCE OR OTHER TORTUOUS ACTIVITY, OR ANY OTHER THEORY OF LIABILITY, EVEN IF CISCO IS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

2014-Mar-06 17:41

Table of Contents

CHAPTER 1
INTRODUCTION

The Sourcefire Event Streamer (also known as eStreamer) allows you to stream Sourcefire 3D System intrusion, discovery, and connection data from the Sourcefire Defense Center or managed device (also referred to as the eStreamer server) to external client applications.

Note that eStreamer is not supported on virtual devices. To stream events from a virtual device, you can configure eStreamer on the Defense Center that the device reports to.

eStreamer uses a custom application layer protocol to communicate with connected client applications. As the purpose of eStreamer is simply to return data that the client requests, the majority of this guide describes the eStreamer formats for the requested data.

There are three major steps to creating and integrating an eStreamer client with a Sourcefire 3D System:

1. Write a client application that exchanges messages with the Defense Center or managed device using the eStreamer application protocol. The eStreamer SDK includes a reference client application.
2. Configure a Defense Center or device to send the required type of events to your client application.
3. Connect your client application to the Defense Center or device and begin exchanging data.

This guide provides the information you need to successfully create and run an eStreamer Version 5.3 client application.

Major Changes in eStreamer Version 5.3

If you are upgrading your Sourcefire 3D System deployment to Version 5.3, please note the following changes, some of which may require you to update your eStreamer client:

- Impact alerts can now handle IPv6 events. See Intrusion Impact Alert Data on page 77 for more information. Added the following data structures:
  - Added IOC State Data Block for 5.3+ on page 158 to provide information on the dynamic analysis of files.
  - Added IOC Name Data Block for 5.3+ on page 160 to provide information about Indications of Compromise (IOCs).
  - Added IOC State Data Block for 5.3+ on page 158 to store information about IOCs.
- Replaced the following blocks:
  - Replaced Full Host Profile Data Block 5.2.x on page 696 with Full Host Profile Data Block 5.3+ on page 388, which has new fields supporting IOC information.
  - Replaced Connection Statistics Data Block 5.2.x on page 602 with Connection Statistics Data Block 5.3+ on page 300, which has fields for NetFlow support.
  - Replaced Malware Event Data Block 5.2.x on page 505 with Malware Event Data Block 5.3+ on page 140, which has new fields supporting IOC information.
  - Replaced File Event for 5.2.x on page 623 with File Event for 5.3+ on page 133, which has new fields supporting IOC information.
  - Replaced Intrusion Event Record 5.2.x on page 478 with Intrusion Event Record 5.3+ on page 70, which has new fields supporting IOC information.

Using this Guide

At the highest level, the eStreamer service is a mechanism for streaming data from the Sourcefire 3D System to a requesting client. The service can stream the following categories of data:

- Intrusion event data and event extra data
- Correlation (compliance) event data
- Discovery event data
- User event data
- Metadata for events
- Host information
- Malware event data

Descriptions of the data structures returned by eStreamer make up the majority of this book. The chapters in the book are:

- Chapter 2, Understanding the eStreamer Application Protocol, which provides an overview of eStreamer communications, details some of the requirements for writing eStreamer client applications, and describes the four types of messages used to send commands to and receive data from the eStreamer service.
- Chapter 3, Understanding Intrusion and Correlation Data Structures, which documents the data formats used to return event data generated by the intrusion detection and correlation components and the data formats used to represent the intrusion and correlation events.
- Chapter 4, Understanding Discovery & Connection Data Structures, which documents the data formats used to return discovery, user, and connection event data.
- Chapter 5, Understanding Host Data Structures, which documents the data formats that eStreamer uses to return full host information data when it receives a host information request message.
- Chapter 6, Configuring eStreamer, which documents how to configure the eStreamer on a Defense Center or managed device. The chapter also documents the eStreamer command-line switches and provides instructions for manually starting and stopping the eStreamer service and for configuring the Defense Center or managed device to start eStreamer automatically.
- Appendix A, Data Structure Examples, which provides examples of eStreamer message packets in binary format.
- Appendix B, Understanding Legacy Data Structures, which documents the structure of legacy data structures that are no longer in use by the currently shipping product but may be used by older clients.

Prerequisites

To understand the information in this guide, you should be familiar with the features and nomenclature of the Sourcefire 3D System and the function of its components in general, and with the different types of event data these components generate in particular. Definitions of unfamiliar or product-specific terms can frequently be obtained from the *Sourcefire 3D System eStreamer Integration Guide*.

Product Versions for Sourcefire 3D System Releases

Version numbers are used throughout this guide to describe the data format for events generated by the Sourcefire Device and Defense Center. The Sourcefire 3D System Product Versions table lists versions for each product by major release.

Sourcefire 3D System Product Versions

RELEASE	DEFENSE CENTER VERSION	MASTER DEFENSE CENTER VERSION	INTRUSION SENSOR VERSION	SENSOR VERSION	SOURCEFIRE DEVICE VERSION
Sourcefire IMS 3.0	Management Console 3.0	N/A	Network Sensor 3.0	N/A	N/A
Sourcefire IMS 3.1	Management Console 3.1	N/A	Network Sensor 3.1	RNA Sensor 1.0	N/A
Sourcefire IMS 3.2	Management Console 3.2	N/A	Network Sensor 3.2	RNA Sensor 2.0	N/A
Sourcefire 3D 4.0	Defense Center 4.0	N/A	Intrusion Sensor 4.0	RNA Sensor 3.0	N/A
Sourcefire 3D 4.5	Defense Center 4.5	N/A	Intrusion Sensor 4.5	RNA Sensor 3.5	N/A
Sourcefire 3D 4.6.1	Defense Center 4.6.1	Master Defense Center 4.6.1	N/A	N/A	Device 4.6.1
Sourcefire 3D 4.7	Defense Center 4.7	Master Defense Center 4.7	N/A	N/A	Device 4.7
Sourcefire 3D 4.8	Defense Center 4.8	Master Defense Center 4.8	N/A	N/A	Device 4.8
Sourcefire 3D 4.8.0.2	Defense Center 4.8.0.2	Master Defense Center 4.8.0.2	N/A	N/A	Device 4.8.0.2
Sourcefire 3D 4.9	Defense Center 4.9	Master Defense Center 4.9	N/A	N/A	Device 4.9

Sourcefire 3D System Product Versions (Continued)

RELEASE	DEFENSE CENTER VERSION	MASTER DEFENSE CENTER VERSION	INTRUSION SENSOR VERSION	SENSOR VERSION	SOURCEFIRE DEVICE VERSION
Sourcefire 3D 4.9.1	Defense Center 4.9.1	Master Defense Center 4.9.1	N/A	N/A	Device 4.9.1
Sourcefire 3D 4.10	Defense Center 4.10	Master Defense Center 4.10	N/A	N/A	Device 4.10
Sourcefire 3D 4.10.1	Defense Center 4.10.1	Master Defense Center 4.10.1	N/A	N/A	Device 4.10.1
Sourcefire 3D 4.10.2	Defense Center 4.10.2	Master Defense Center 4.10.2	N/A	N/A	Device 4.10.2
Sourcefire 3D 5.0	Defense Center 5.0	N/A	N/A	N/A	Device 5.0
Sourcefire 3D 5.1	Defense Center 5.1	N/A	N/A	N/A	Device 5.1
Sourcefire 3D 5.1.1	Defense Center 5.1.1	N/A	N/A	N/A	Device 5.1.1
Sourcefire 3D 5.2	Defense Center 5.2	N/A	N/A	N/A	Device 5.2
Sourcefire 3D 5.3	Defense Center 5.3	N/A	N/A	N/A	Device 5.3

Document Conventions

The eStreamer Message Data Type Conventions table lists the names used in this book to describe the various data field formats employed in eStreamer messages. Numeric constants used by the eStreamer service are typically unsigned integer values. Bit fields use low-order bits unless otherwise noted. For example, in a one byte field containing five bits of flag data, the low-order five bits will contain the data.

eStreamer Message Data Type Conventions

DATA TYPE	DESCRIPTION
nn-bit field	Bit field of nn bits
byte	8-bit byte containing data of arbitrary format
int8	Signed 8-bit byte
uint8	Unsigned 8-bit byte
int16	Signed 16-bit integer
uint16	Unsigned 16-bit integer
int32	Signed 32-bit integer
uint32	Unsigned 32-bit integer
uint64	Unsigned 64-bit integer
string	Variable length field containing character data
[n]	Array subscript following any of the above data types to indicate n instances of the indicated data type, for example, uint8[4]
variable	Collection of various data types
BLOB	Binary object of unspecified type, typically raw data as captured from a packet

IP Addresses

The Sourcefire database stores IPv4 and IPv6 addresses in the same fields in a BINARY format. To get IPv6 addresses, convert to hex notation, for example: `20010db8000000000000000000004321`. The database follows the RFC for storing IPv4 addresses by filling in bits 80-95 with 1's, which yields an invalid IPv6 address. For example, the IPv4 address 10.5.15.1 would be stored as `00000000000000000000FFFF0A050F01`.

CHAPTER 2

UNDERSTANDING THE ESTREAMER APPLICATION PROTOCOL

The Sourcefire Event Streamer (eStreamer) uses a message-oriented protocol to stream events and host profile information to your client application. Your client can request event and host profile data from a Defense Center, and intrusion event data only from a managed device. Your client application initiates the data stream by submitting request messages, which specify the data to be sent, and then controls the message flow from the Defense Center or managed device once streaming begins.

Throughout this document, the eStreamer service on the Defense Center or a managed device may be referred to as the eStreamer server or eStreamer.

The following sections describe requirements for connecting to the eStreamer service and introduce commands and data formats used in the eStreamer protocol:

- Connection Specifications on page 17 describes the communication flow between the eStreamer service and your client and describes how the client interacts with it.
- Understanding eStreamer Communication Stages on page 17 describes the communication protocol for client applications to submit data requests to the eStreamer server and for eStreamer to deliver the requested information to the client.
- Understanding eStreamer Message Types on page 22 describes the message types used in the eStreamer protocol, discusses the basic structure of data packets used by eStreamer to return intrusion event data, discovery event data, metadata, and host data to a client, and provides other information to help you write a client that can interpret eStreamer messages.

Connection Specifications

The eStreamer service:

- Communicates using TCP over an SSL connection (the client application must support SSL-based authentication).
- Accepts connection requests on port 8302.
- Waits for the client to initiate all communication sessions.
- Writes all message fields in network byte order (big endian).
- Encodes text in UTF-8.

Understanding eStreamer Communication Stages

There are four major stages of communication that occur between a client and the eStreamer service:

1. The client establishes a connection with the eStreamer server and the connection is authenticated by both parties.

   See Establishing an Authenticated Connection on page 18 for more information.

2. The client requests data from the eStreamer service and specifies the types of data to be streamed. A single event request message can specify any combination of available event data, including event metadata. A single host profile request can specify a single host or multiple hosts.

   Two request modes are available for requesting event data:

   - Event Stream Request - the client submits a message containing request flags that specify the requested event types and version of each type, and the eStreamer server responds by streaming the requested data.
   - Extended Request - the client submits a request with the same message format as for Event Stream requests but sets a flag for an extended request. This initiates a message interaction between client and eStreamer server through which the client requests additional information and version combinations not available via Event Stream requests.

   For information on requesting data, see Requesting Data from eStreamer on page 19.

3. eStreamer establishes the requested data stream to the client.

   See Accepting Data from eStreamer on page 21 for more information.

4. The connection terminates.

   See Terminating Connections on page 22 for more information.

Establishing an Authenticated Connection

Before a client can request data from eStreamer, the client must initiate an SSL-enabled TCP connection with the eStreamer service. When the client initiates the connection, the eStreamer server responds, initiating an SSL handshake with the client. As part of the SSL handshake, the eStreamer server requests the client's authentication certificate, and verifies that the certificate is valid (signed by the Internal Certifying Authority [Internal CA] on the eStreamer server).

IMPORTANT! Sourcefire recommends that you also require your client to verify that the certificate presented by the eStreamer server has been signed by a trusted Certifying Authority. This is the Internal CA certificate included in the PKCS#12 file that Sourcefire provides when you register a new eStreamer client with the Defense Center or managed device. See Adding Authentication for eStreamer Clients on page 407 for more information.

After the SSL session is established, the eStreamer server performs an additional post-connection verification of the certificate. This includes verifying that the client connection originates from the host specified in the certificate and that the subject name of the certificate contains the appropriate value. If either post-connection check fails, the eStreamer server closes the connection. If necessary, you can configure the eStreamer service so that it does not perform a client host name check (see eStreamer Service Options on page 413 for more information).

While the client is not required to perform post-connection verification, Sourcefire recommends that the client perform this verification step. The authentication certificate contains the following field values in the subject name of the certificate:

Certificate Subject Name Fields

FIELD	VALUE
`title`	`estreamer`
`generationQualifier`	`server`

After the post-connection verification is finished, the eStreamer server awaits a data request from the client.

Requesting Data from eStreamer

Your client performs the following high-level tasks in managing data requests:

- initializing the request session — see Establishing a Session on page 19.
- requesting events from the eStreamer event archive — Using Event Stream Requests and Extended Requests to Initiate Event Streaming on page 19.
- requesting host data — see Requesting Host Data on page 21.
- changing a request — see Changing a Request on page 21.

Establishing a Session

The client establishes a session by sending an initial Event Stream request to the eStreamer service.

In this initial message, you can either include data request flags or submit the data requests in a follow-on message. This initial Event Stream request message itself is a prerequisite for all eStreamer requests, whether for event data or for host data. For information about using the Event Stream request message, see Event Stream Request Message Format on page 28.

Using Event Stream Requests and Extended Requests to Initiate Event Streaming

The eStreamer service provides two modes of requests for event streaming. Your request can combine modes. In both modes, your client starts the request with an Event Stream request message but sets the request flag bits differently. For details about the Event Stream message format, see Event Stream Request Message Format on page 28.

When eStreamer receives an Event Stream request message, it processes the client request as follows:

- If the request message does **not** set bit 30 in the request flag field, eStreamer begins streaming any events requested by other set bits in the request flag field. For information, see Submitting Event Stream Requests below.
- If bit 30 **is** set in the Event Stream request, eStreamer provides extended request processing. Extended request flags must be sent if this bit is set. For information, see Submitting Extended Requests below. Note that eStreamer resolves any duplicate requests. If you request multiple versions of the same data, either by multiple flags or multiple extended requests, the highest version is used. For example, if eStreamer receives flag requests for discovery events version 1 and 6 and an extended request for version 3, it sends version 6.

Submitting Event Stream Requests

Event stream requests use a simple process:

- Your client sends a request message to the eStreamer service with a start date and time and a request flag field that specifies the events and their version level to be included in the data stream.
- eStreamer streams events beginning at the specified time. For information about the streaming protocol, see Accepting Data from eStreamer on page 21.

For information on the format and content of the client's Event Stream request message, see Event Stream Request Message Format on page 28.

For information on the event types and versions of events that the client can request, see the Request Flags table on page 31.

Submitting Extended Requests

If you set bit 30 in the request flags field of an Event Stream Request message, you initiate an extended request, which starts a negotiation with the server. Extended request flags must be sent if this bit is set. For the event types available by extended request, see the Event Types and Versions for Extended Request table on page 58.

The steps for extended requests are as follows:

- Your client sends an Event Streaming Request message to eStreamer with the request flags bit 30 set to 1, which signals an extended request. See Event Stream Request Message Format on page 28 for message format details.
- eStreamer answers with a Streaming Information message that advertises the list of services available to the client. For details about the Streaming Information message, see Streaming Information Message Format on page 52.
- The client returns a Streaming Request message that indicates the service it wants to use, with a request list of event types and versions available from that service. The request list corresponds to setting bits in the request flag field when making a standard event stream request. For details about how to use the Streaming Request message to request events, see Sample Extended Request Messages on page 60.
- eStreamer processes the client's Streaming Request message and begins streaming the data at the time specified in the message. For information about the streaming protocol, see Accepting Data from eStreamer on page 21.

Requesting Host Data

Once you have established a session, you can submit a request for host data at any time. eStreamer generates information for the requested hosts from the Sourcefire 3D System network map.

Changing a Request

To change request parameters for an established session, the client must disconnect and request a new session.

Accepting Data from eStreamer

IMPORTANT! The eStreamer server does not keep a history of the events it sends. Your client application must check for duplicate events, which can inadvertently occur for a number of reasons. For example, when starting up a new streaming session, the time specified by the client as the starting point for the new session can have multiple messages, some of which may have been sent in the previous session and some of which were not. eStreamer sends all message that meet the specified request criteria. Your application should detect any resulting duplicates.

During periods of inactivity, eStreamer sends periodic null messages to the client to keep the connection open. If it receives an error message from the client or an intermediate host, it closes the connection.

eStreamer transmits requested data to the client differently, depending on the request mode.

Event Stream Requests

If the client submits an event stream request, eStreamer returns data message by message. It may send multiple messages in a row without waiting for a client acknowledgment. At a certain point, it pauses and waits for the client. The client operating system buffers received data and lets the client process it at its own pace.

If the client request includes a request for metadata, eStreamer sends the metadata first. The client should store it in memory to be available when processing the event records that follow.

Extended Requests

If the client submits an extended request, eStreamer queues up messages and sends them in bundles. eStreamer may send multiple bundles in a row without waiting for a client acknowledgment. At a certain point, it pauses and waits for the client. The client operating system buffers received data and lets the client read it off at its own pace.

The client unpacks each bundle, message by message, and uses the lengths of the records and the blocks to parse each message. The overall message length in each message header can be used to calculate when the end of each message has been reached, and the overall bundle length can be used to know when the end of the bundle is reached. The bundle requires no index of its contents to be correctly parsed.

For information about the message bundling mechanism, see Message Bundle Format on page 61.

For information about the null message that the client can use for additional flow control, see Null Message Format on page 25.

Terminating Connections

The eStreamer server attempts to send an error message before closing the connection. For information on error messages, see Error Message Format on page 26.

The eStreamer server can close a client connection for the following reasons:

- Any time sending a message results in an error. This includes both event data messages and the null keep-alive message eStreamer sends during periods of inactivity.
- An error occurs while processing a client request.
- Client authentication fails (no error message is sent).
- eStreamer service is shutting down (no error message is sent).

Your client can close the connection to eStreamer server at any time and should attempt to use the error message format to notify the eStreamer server of the reason. For information, see Error Message Format on page 26.

Understanding eStreamer Message Types

The eStreamer application protocol uses a simple message format that includes a standard message header and various sub-header fields followed by the record data which contains the message’s payload. The message header is the same in all eStreamer message types; for more information, see eStreamer Message Header on page 24.

The eStreamer Message Types table describes the available message types.

eStreamer Message Types

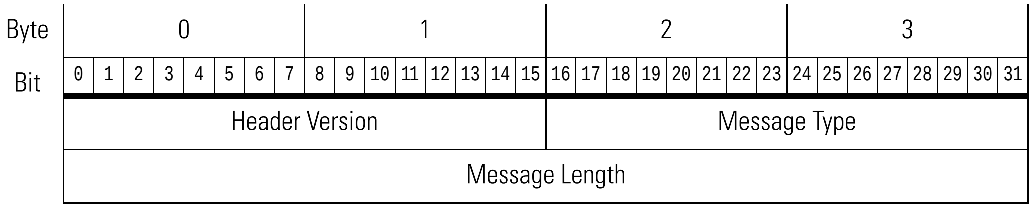
Message Type	Name	Description
0	Null message	Both the eStreamer server and the client send null messages to control data flow. For information, see Null Message Format on page 25.
1	Error message	Both the eStreamer server and the client use error messages to indicate why a connection closed. For information, see Error Message Format on page 26.
2	Event Stream Request	A client sends this message type to the eStreamer service to initiate a new streaming session and request data. For information, see Event Stream Request Message Format on page 28.
4	Event Data	The eStreamer service uses this message type to send event data and metadata to the client. For information, see Event Data Message Format on page 37.
5	Host Data Request	A client sends this message type to the eStreamer service to request host data. A session must be started already via an Event Stream Request message. For information, see Host Request Message Format on page 47.
6	Single Host Data	The eStreamer service uses this message type to send single host data requested by the client. For information, see Host Data and Multiple Host Data Message Format on page 51.
7	Multiple Host Data	The eStreamer service uses this message type to send multiple host data requested by the client. For information, see Host Data and Multiple Host Data Message Format on page 51.

eStreamer Message Types (Continued)

Message Type	Name	Description
2049	Streaming Request	A client uses this message type in extended requests to specify which of the advertised events from the Stream Information message it wants. For information, see Sample Extended Request Messages on page 60.
2051	Streaming Information	The eStreamer service uses this message type in extended requests to advertise the list of services available to the client. For information, see Streaming Information Message Format on page 52.
4002	Message Bundle	The eStreamer service uses this message type to package messages that it streams to clients. For information, see Message Bundle Format on page 61.

eStreamer Message Header

All eStreamer messages start with the message header illustrated in the graphic below. The following table explains the fields.



Standard eStreamer Message Header Fields

Field	Data Type	Description
Header Version	uint16	Indicates the version of the header used on the message. For the current version of eStreamer, this value is always 1.

Standard eStreamer Message Header Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Message Type	uint16	Indicates the type of message transmitted. For the list of current values, see the eStreamer Message Types table on page 23.
Message Length	uint32	Indicates the length of the content that follows, and excludes the bytes in the message header itself. A message with a header and no data has a message length of zero.

Null Message Format

Both the client application and the eStreamer service send null messages. The null message has a type of 0 and contains no data after the message header.

The client sends a null message to the eStreamer server to indicate readiness to accept more data. The eStreamer service sends null messages to the client to keep the connection alive when no data is being transmitted. The message length value for null messages is always set to 0.

TIP! In data structure diagrams in this book, integers in parentheses such as (1) or (115) represent constant field values. For example, Header Version (1) means that the field in the data structure under discussion always has a value of 1.

The Null message format is shown below. The only non-zero value in the message is the header version.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (0)</td></tr>
<tr><td></td><td colspan="32">Message Length (0)</td></tr>
</table>

An example of a null message in binary format follows. Notice that the only non-zero value is in the second byte, signifying a header version value of one. The message type and length fields (shaded) each have a value of zero.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIP! Examples in this guide appear in binary format to clearly display which bits are set. This is important for some messages, such as the event request message and event impact fields.

Error Message Format

Both the client application and the eStreamer service use error messages. Error messages have a message type of 1 and contain a header, an error code, an error text length, and the actual error text. Error text can contain between zero and 65,535 bytes.

When you create custom error messages for your client application, Sourcefire recommends using -1 as the error code.

The following graphic illustrates the basic error message format. Shaded fields are specific to error messages.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (1)															
	Message Length																															
	Error Code																															
	Error Text Length																Error Text...															

The Error Message Fields table describes each field in error code messages.

Error Message Fields

Field	Data Type	Description
Error Code	int32	A number representing the error.
Error Text Length	uint16	The number of bytes included in the error text field.
Error Text	variable	The error message. Up to 65,535 bytes.

The following diagram shows an example error message:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
D	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	1	0	0	1	1	0	1	1	1	1
	0	0	1	0	0	0	0	0	0	1	1	1	0	0	1	1	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1
	0	1	1	0	0	0	1	1	0	1	1	0	0	1	0	1																

In the preceding example, the following information appears:

A. The first two bytes indicate the standard header value of 1. The second two bytes show a value of 1, which signifies that the transmission is an error message.

B. This line indicates the amount of message data that follows it. In this example, 15 bytes (in binary, 1111) of data follow.

C. This line displays the error code. In this example, the message contains a value of 19 (10011). Therefore, error number 19 is transmitted in the message.

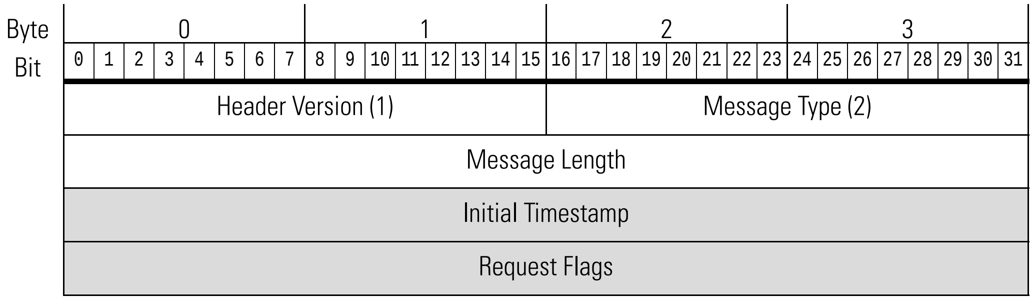
D. This line contains the number of bytes in the error message (1001, or nine bytes), and the error message itself follows in the next nine bytes. The error message value, when converted to ASCII text, equals “No space,” which is the error message that accompanies error code 19.

Event Stream Request Message Format

eStreamer clients use the Event Stream Request message to start a streaming session. The request message includes a start time and a bit flag field to specify the data the eStreamer service should include, which can be any combination of events, as well as intrusion event extra data and metadata. The Event Stream Request message can initiate both event stream requests and extended requests. The message type is 2.

You must submit an Event Stream Request message for all data requests, including a request exclusively for host profile information. In such a case, you first submit an Event Stream Request message, then a Host Request message (type 5) to specify the host data.

The following graphic illustrates the Event Stream Request message format. The message uses the standard header. The shaded fields are specific to the request message and are described in the following table.



The Event Stream Request Message Fields table describes each field in Event Stream Request messages.

Event Stream Request Message Fields

FIELD	DATA TYPE	DESCRIPTION
Initial Timestamp	uint32	Defines the start of the session. To start at: • the time the client connects to eStreamer, set all timestamp bits to 1. • the oldest data available, set all timestamp bits to zero. • a given date and time, specify the UNIX timestamp (number of seconds since January 1, 1970). See Initial Timestamp below for important information.
Request Flags	bits[32]	Specifies the types and versions of events and metadata to be returned in event stream requests. See Request Flags on page 30 for flag definitions. Setting bit 30 initiates an extended request, which can co-exist with event stream requests in the same message.

Initial Timestamp

IMPORTANT! Your client application should use the archival timestamp in the Initial Timestamp field when submitting an event stream request, as explained below. This ensures that you do not inadvertently exclude events. Devices transmit data to the Defense Center using a “store and forward” mechanism with transmission delays. If you request events by the generation timestamp assigned by the device that detects it, delayed events may be missed.

When starting a session, a best practice is to start up from the archival timestamp (also known as the “server timestamp”) of the last record in the previous session. It is not a technical requirement but is strongly recommended. Under certain circumstances, if you use the generation timestamp you can inadvertently exclude events from the new streaming session.

To include the archival timestamp in your streamed events, you must set bit 23 in the request flag field.

Note that only time-based events have archival timestamps. Events that eStreamer generates, such as metadata, have zero in this field when extended event headers have been requested with bit 23 set.

Request Flags

You set bits 0 through 29 in the event data request flag field to select the types of events you want eStreamer to send. You set bit 30 to activate the extended request mode. Setting bit 30 does not directly request any data. Extended request flags must be sent if this bit is set. Your client requests data during the server-client message dialog that follows submission of the Event Stream Request message. For information on extended requests, see Requesting Data from eStreamer on page 19.

See the Request Flags table on page 31 for definitions of the bit settings in the Request Flags field. Different flags request different versions of the event data. For example, to obtain data in Sourcefire 3D 4.9 format instead of 4.10 format you set a different flag bit. For specific information on the flags to use when requesting data for particular product versions, see the Event Request Flags by Product Version table on page 35.

Note that you request metadata by version, not by the individual metadata record. For information about each supported version of metadata, see Request Flags on page 30.

The following diagram shades the bits in the flags field that are currently used:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
Flag Bit		30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

For information on each request flag bit, see the Request Flags table that follows.

Request Flags

Bit Field	Description
Bit 0	Requests the transmission of packet data associated with intrusion events. If set to 1, packet data is transmitted with intrusion events. If set to 0, packet data is not transmitted.
Bit 1	Requests the transmission of version 1 metadata associated with intrusion, discovery, correlation, and connection events. If set to 1, version 1 metadata is transmitted with events. If set to 0, version 1 metadata is not transmitted. You can use metadata to resolve coded and numeric fields in events. See Understanding Metadata on page 63 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 2	Requests the transmission of intrusion events. If bit 2, bit 6, or both bit 2 and 6 are set to `1`, but the extended request flag, bit 30, is set to `0`, the system interprets this as a request from a Version 4.x client and record type 104/105 is sent. If no event type is specified when bit 2, bit 6, or both bit 2 and 6 are set to `1`, and bit 30 is set to `1`, the system interprets this as a request from a Version 5.0-5.1 client and record type 207/208 is sent. If bit 30 is set to `1`, and a specific event type is requested, intrusion events are sent regardless of bits 2 and 6. For details on requesting record types, see Submitting Extended Requests on page 20. If bit 2, bit 6, and bit 30 are all set to `0`, intrusion events are not sent. Bit 6 is used in a manner identical to bit 2. Either bit can be set to request intrusion events. Setting one of these bits to `0` will not override the other bit; setting bit 2 to `0` and bit 6 to `1`, or setting bit 2 to `1` and bit 6 to `0`, will be interpreted as a request for intrusion events.
Bit 3	Requests the transmission of discovery data version 1 (Defense Center 3.2). If set to 0, discovery data version 1 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures on page 164.
Bit 4	Requests the transmission of correlation data version 1 (Defense Center 3.2). If set to 0, correlation data version 1 is not transmitted.
Bit 5	Requests the transmission of impact correlation events (intrusion impact alerts). If set to 1, intrusion impact alerts are transmitted. If set to 0, intrusion impact alerts are not transmitted. See Intrusion Impact Alert Data on page 77 for more information about intrusion impact alerts.
Bit 6	Bit 6 is used in a manner identical to bit 2. See Bit 2 on page 31.
Bit 7	Requests the transmission of discovery data version 2 (Defense Center 4.0 - 4.1) if set to 1. If set to 0, discovery data version 2 is not transmitted.

Request Flags (Continued)

Bit Field	Description
Bit 8	Requests the transmission of connection data version 1 (Defense Center 4.0 - 4.1) if set to 1. If set to 0, connection data version 1 is not sent.
Bit 9	Requests the transmission of correlation data version 2 (Defense Center 4.0 - 4.1.x) if set to 1. If set to 0, correlation policy data version 2 is not transmitted.
Bit 10	Requests the transmission of discovery data version 3 (Defense Center 4.5 - 4.6.1) if set to 1. If set to 0, discovery data version 3 is not transmitted. For more information about legacy discovery events, see Legacy Discovery Data Structures on page 513.
Bit 11	Disables transmission of events.
Bit 12	Requests the transmission of connection data version 3 (Defense Center 4.5 - 4.6.1) if set to 1. If set to 0, connection data version 3 is not sent.
Bit 13	Requests the transmission of correlation data version 3 (Defense Center 4.5 - 4.6.1). If set to 0, correlation data version 3 is not transmitted.
Bit 14	Requests the transmission of version 2 metadata associated with intrusion, discovery, correlation, and connection events. If set to 1, version 2 metadata is transmitted with events. If set to 0, version 2 metadata is not transmitted. See Understanding Metadata on page 63 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 15	Requests the transmission of version 3 metadata associated with intrusion, correlation, discovery, and connection events. If set to 1, version 3 metadata is transmitted with events. If set to 0, version 3 metadata is not transmitted. See Understanding Metadata on page 63 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 16	Unused
Bit 17	Requests the transmission of discovery data version 4 (Defense Center 4.7 - 4.8.x). If set to 0, discovery data version 4 is not transmitted.
Bit 18	Requests the transmission of connection data version 4 (Defense Center 4.7 - 4.9.0.x) if set to 1. If set to 0, connection data version 4 is not sent. See Connection Statistics Data Block for 4.7 - 4.9.0.x on page 577 and Connection Chunk Message on page 216 for more information.

Request Flags (Continued)

Bit Field	Description
Bit 19	Requests the transmission of correlation data version 4 (Defense Center 4.7). If set to 0, correlation data version 4 is not transmitted. See Legacy Correlation Event Data Structures on page 630 for information about correlation events transmitted in Defense Center 4.7 format.
Bit 20	Requests the transmission of version 4 metadata associated with intrusion, discovery, user activity, correlation, and connection events. If set to 1, version 4 metadata is transmitted with events. If set to 0, version 4 metadata is not transmitted. Version 4 metadata includes the following: • correlation (compliance) rule information • correlation (compliance) policy information • fingerprint records • client application records • client application type records • vulnerability records • host criticality records • network protocol records • host attribute records • scan type records • user records • service detection device (version 2) records • event classification (version 2) records • priority records • rule information (version 2) • malware information If you request bit 20 with bit 22, user metadata is also sent. See Understanding Metadata on page 63 for general information on the way eStreamer transmits metadata to clients and how a client can use metadata.
Bit 21	Requests the transmission of version 1 user events. For more information on user events, see User Record on page 188.
Bit 22	Requests the transmission of correlation data version 5 (Defense Center 4.8.0.2 - 4.9.1). If set to 0, correlation data version 5 is not transmitted. If you request bit 20 with bit 22, user metadata is also sent. For more information about legacy correlation (compliance) events, see Legacy Correlation Event Data Structures on page 630.

Request Flags (Continued)

Bit Field	Description
Bit 23	Requests extended event headers. If set to 1, events are transmitted with the timestamp applied when the event was archived for the eStreamer server to process and four bytes reserved for future use. If this field is set to 0, events are sent with a standard event header that only includes the record type and record length. See eStreamer Message Header on page 24 for information about the event message header.
Bit 24	Requests the transmission of discovery data version 5 (Defense Center 4.9.0.x). If set to 0, discovery data version 5 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures on page 164.
Bit 25	Requests the transmission of discovery data version 6 (Defense Center 4.9.1+). If set to 0, discovery data version 6 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures on page 164.
Bit 26	Requests the transmission of connection data version 5 (Defense Center 4.9.1 - 4.10.x) if set to 1. If set to 0, connection data version 5 is not sent. See Connection Statistics Data Block 4.9.1 - 4.10.1 on page 581 and Connection Chunk Message on page 216 for more information.
Bit 27	Requests event extra data associated with an intrusion event in an Extra Data record. For more information about event data, see Intrusion Event Extra Data Data Block Fields on page 90.
Bit 28	Requests the transmission of discovery data version 7 (Defense Center 4.10.0+). If set to 0, discovery data version 7 is not transmitted. For more information about discovery events, see Understanding Discovery & Connection Data Structures on page 164.
Bit 29	Requests the transmission of correlation data version 6 (Defense Center 4.10 - 4.10.x). If set to 0, correlation policy data version 6 is not transmitted. If you request bit 20 with bit 29, user metadata is also sent. For more information about correlation events, see Correlation Event for 4.10.x on page 638.
Bit 30	Indicates an extended request to eStreamer. Extended request flags must be sent if this bit is set. For information about extended requests, see Submitting Extended Requests on page 20.

To help you decide which flags to use to request data for a particular version, see the Event Request Flags by Product Version table that follows.

Event Request Flags by Product Version

Type of Requested Data	3D/DC 4.9.0.x	3D/DC 4.9.1.x	3D/DC 4.10.x	3D/DC 5.0+	3D/DC 5.1	3D/DC 5.1.1+
packet data	Bit 0	Bit 0	Bit 0	Bit 0	Bit 0	Bit 0
intrusion events	Bit 2	Bit 2	Bit 2	Bit 2	Bit 2	Bit 30 (see Submitting Extended Requests on page 20)
metadata	Bit 20	Bit 20	Bit 20	Bit 20	Bit 20	Bit 20
discovery events	Bit 24	Bit 25	Bit 28	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)
correlation events	Bit 22	Bit 22	Bit 29	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)
event extra data			Bit 27	Bit 27	Bit 27	Bit 27
impact event alerts	Bit 5	Bit 5	Bit 5	Bit 5	Bit 5	Bit 5
connection data	Bit 18	Bit 26	Bit 26	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)
user events	Bit 21	Bit 21	Bit 21	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)	Bit 30 (see Submitting Extended Requests on page 20)

Event Request Flags by Product Version (Continued)

Type of Requested Data	3D/DC 4.9.0.x	3D/DC 4.9.1.x	3D/DC 4.10.x	3D/DC 5.0+	3D/DC 5.1	3D/DC 5.1.1+
malware events						Bit 30 (see Submitting Extended Requests on page 20)
file events						Bit 30 (see Submitting Extended Requests on page 20)

WARNING! In all event types, prior to version 5.x, the reference client labels `detection engine ID` fields as `sensor ID`.

The following example requests intrusion events of type 7 (compatible with Sourcefire 3D 3.2+) with both version 1 metadata and packet flags:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
Flag Bit		30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

To request only data compatible with Sourcefire 3D 3.2 (including intrusion events, packets, metadata, impact alerts, policy violation events, and version 2.0 events), use the following:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
Flag Bit		30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

To request intrusion impact alerts, correlation events, discovery events, connection events, and intrusion events of type 7 with packets and version 3 metadata in Defense Center 4.6.1+ format, use the following:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	0	1	0	1
Flag Bit		30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Event Data Message Format

The eStreamer service transmits event data and related metadata to clients when it receives an event request. Event data messages have a message type of 3. Each message contains a single data record with either event data or metadata.

Note that type 3 messages carry only event data and metadata. eStreamer transmits host information in type 6 (single-host) and type 7 (multiple-host) messages. See Host Data and Multiple Host Data Message Format on page 51 for information on host message formats.

Understanding the Organization of Event Data Messages

The event data and metadata messages that eStreamer sends contain the following sections:

- eStreamer message header — the standard message header defined at eStreamer Message Header on page 24.
- Event-specific sub-headers — sets of fields that vary by event type, with codes that describe additional event details and determine the structure of the payload data that follows.
- Data record — fixed-length fields and a data block.

IMPORTANT! The client should unpack all messages on the basis of field length.

For the event message formats by event type, see the following:

- Intrusion Event and Metadata Message Format on page 39 — for intrusion event data records and all metadata records. These messages have fixed-length fields.
- Discovery Event Message Format on page 40 — for messages with discovery event or user event data. In addition to the standard eStreamer message header and a record header similar to the intrusion event message, discovery messages have a distinctive discovery event header with an event type and subtype field. The data record in discovery event messages is packaged in a series 1 block that can have variable length fields and multiple layers of encapsulated blocks.
- Connection Event Message Format on page 42 — for messages with connection statistics. Their general structure is identical to discovery event messages. Their data block types, however, are specific for connection statistics.
- Correlation Event Message Format on page 42 — for messages with correlation (compliance) event data. The headers in these messages are the same as in intrusion event messages but the data blocks are series 1 blocks.
- Event Extra Data Message Format on page 44 — for a series of messages that deliver intrusion-related record types with variable-length fields and multiple layers of nested data blocks such as intrusion event extra data. See Event Extra Data Message Format on page 44 for general information on the structure of this message series. See Data Block Header on page 46 for information about the structures of this series of blocks which are similar to series 1 blocks but numbered separately.

Intrusion Event and Metadata Message Format

The graphic below shows the general structure of intrusion event and metadata messages.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Message Header See eStreamer Message Header on page 24			
	Record Header			
	Data Record ...			

The following graphic shows the details of the record header portion of the intrusion event and metadata message format. The record header fields are shaded. The table that follows defines the fields.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (3)	
	Message Length			
	Record Type See Intrusion Event and General Metadata Record Types on page 65			
	Record Length			
	eStreamer Server Timestamp (for events only, not used in metadata records)			
	Reserved for Future Use (for events only, not used in metadata records)			
	Data ...			

The Intrusion Event and Metadata Record Header Fields table describes each field in the header of intrusion events and metadata messages.

Intrusion Event and Metadata Record Header Fields

FIELD	DATA TYPE	DESCRIPTION
Record Type	uint32	Identifies the data record content type. See the Intrusion Event and General Metadata Record Types table on page 65 for the list of record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request message flags.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags.

Discovery Event Message Format

The graphic below shows the structure of discovery event messages. The standard eStreamer message header and event record header are followed by a discovery event header used only in discovery and user event messages. The discovery event header section of the message contains the discovery event type and subtype fields, which together form a key to the data block that follows. For the current discovery event types and subtypes, see the Discovery and Connection Events by Type and Subtype table on page 201.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Message Header
See eStreamer Message Header on page 24</td></tr>
<tr><td></td><td colspan="32">Discovery Event Record Header
See Discovery Event Message Headers on page 41 for field details.</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header
See Discovery Event Header 5.2+ on page 198 for field details.</td></tr>
<tr><td></td><td colspan="32">Series 1 Data Block
See Understanding Discovery (Series 1) Blocks on page 224
...</td></tr>
</table>

Discovery Event Message Headers

The shaded section in the following graphic shows the fields of the record header in the discovery event data message format, and shows the location of the event header that follows it. The table below defines the fields of the discovery event message headers.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (3)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type
See Discovery and Connection Event Record Types on page 166</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">eStreamer Server Timestamp
(for events only)</td></tr>
<tr><td></td><td colspan="32">Reserved for Future Use
(for events only)</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header
See Discovery Event Header Fields on page 200</td></tr>
<tr><td></td><td colspan="32">Series 1 Data Block
See Understanding Discovery (Series 1) Blocks on page 224
...</td></tr>
</table>

The Discovery Event Message Header Fields table describes the fields in the record header and the event header of the discovery event message.

Discovery Event Message Header Fields

FIELD	DATA TYPE	DESCRIPTION
Record Type	uint32	Identifies the data record content type. See the Discovery and Connection Event Record Types table on page 166 for the list of record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request flags field of the event stream request.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags.
Discovery Event Header	Varied	Contains a number of fields, including the event type and subtype, which together form a unique key to the data structure that follows. See Discovery Event Header 5.2+ on page 198 for definitions of fields in the discovery event header.

Connection Event Message Format

Messages with connection statistics have a structure identical to discovery event messages. See Discovery Event Message Format on page 40 for general message format information. Connection event messages are distinct in terms of the data block types they incorporate.

Correlation Event Message Format

The graphic below shows the general structure of correlation (compliance) event messages. The standard eStreamer message header and record header are followed immediately by a data block in the data record section of the message. Correlation messages use Series 1 data blocks.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Message Header
See eStreamer Message Header on page 24</td></tr>
<tr><td></td><td colspan="32">Record Header
See Correlation Record Header on page 43 for field details.</td></tr>
<tr><td></td><td colspan="32">Data Record Block
...</td></tr>
</table>

Correlation Record Header

The shaded section of the following graphic shows the fields of the record header in correlation event messages. Note that correlation messages use series 1 data blocks; however, they do not have the discovery header that appears in discovery event messages. Their header fields resemble those of intrusion event messages. The table that follows the graphic below defines the record header fields for correlation events.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (3)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type
See Intrusion Event and General Metadata Record Types on page 65</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">eStreamer Server Timestamp
(for events only, not used in metadata records)</td></tr>
<tr><td></td><td colspan="32">Reserved for Future Use
(for events only, not used in metadata records)</td></tr>
<tr><td></td><td colspan="32">Data Record Block
Uses Series 1 block, see Understanding Discovery (Series 1) Blocks on page 224
...</td></tr>
</table>

The Correlation Event Message Record Header Fields table describes each field in the record header of correlation event messages.

Correlation Event Message Record Header Fields

FIELD	DATA TYPE	DESCRIPTION
Record Type	uint32	Identifies the data record content type. See the Intrusion Event and General Metadata Record Types table on page 65 for the list of intrusion, correlation, and metadata record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request message flags. Field is zero for data generated by the Defense Center such as host profiles and metadata.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags.

Event Extra Data Message Format

The graphic below shows the structure of event extra data messages. The Intrusion Event Extra Data message is an example of this message group.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Message Header
See eStreamer Message Header on page 24</td></tr>
<tr><td></td><td colspan="32">Record Header
See Event Extra Data Message Record Header on page 45</td></tr>
<tr><td></td><td colspan="32">Data Blocks...
See Understanding Series 2 Data Blocks on page 116</td></tr>
</table>

Event extra data messages have the same format as correlation event messages, with a data block directly after the record header. Unlike correlation messages, they use series 2 data blocks, not series 1 data blocks, which have a separate numbering sequence. For information about series 2 block types, see Understanding Series 2 Data Blocks on page 116.

Event Extra Data Message Record Header

The shaded section of the following graphic shows the fields of the record header in event extra data messages. The table that follows defines the record header fields for event extra data messages.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (3)	
	Message Length			
	Record Type See Intrusion Event and General Metadata Record Types on page 65			
	Record Length			
	eStreamer Server Timestamp (for events only, not used in metadata records)			
	Reserved for Future Use (for events only, not used in metadata records)			
	Data Record Block Uses series 2 block, see Understanding Series 2 Data Blocks on page 116 ...			

The Event Extra Data Message Record Header Fields table describes each field in the record header of event extra data messages.

Event Extra Data Message Record Header Fields

FIELD	DATA TYPE	DESCRIPTION
Record Type	uint32	Identifies the data record content type. See the Intrusion Event and General Metadata Record Types table on page 65 for the list of event extra data record types.
Record Length	uint32	Length of the content of the message after the record header. Does not include the 8 or 16 bytes of the record header. (Record Length plus the length of the record header equals Message Length.)

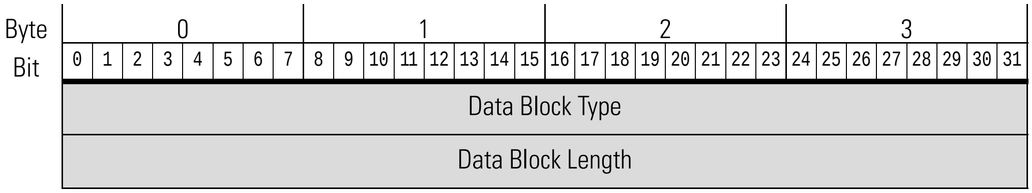
Event Extra Data Message Record Header Fields (Continued)

Field	Data Type	Description
eStreamer Server Timestamp	uint32	Indicates the timestamp applied when the event was archived by the eStreamer server. Also called the archival timestamp. Field present only if bit 23 is set in the request message flags. Field is not present for events generated by the Defense Center.
Reserved for future use	uint32	Reserved for future use. Field present only if bit 23 is set in the request message flags. Field is not present for events generated by the Defense Center.

Data Block Header

Series 1 blocks and series 2 blocks have similar structures but distinct numbering. These blocks can appear anywhere in the data portion of a discovery, correlation, connection, or event extra data message. These blocks encapsulate other blocks at multiple levels of nesting.

The data blocks in both the first and second series begin with the header structure shown in the graphic below. The following table provides information about the header fields. The header is followed immediately by the data structure associated with the data block type.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Data Block Type			
	Data Block Length			

Field	Data Type	Description
Data Block Type	uint32	For series 1 block types, see Understanding Discovery (Series 1) Blocks on page 224. For series 2 block types, see the Series 2 Block Types table on page 117.
Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.

Host Request Message Format

To receive host profiles, you submit Host Request messages. You can request data for a single host or multiple hosts defined by an IP address range.

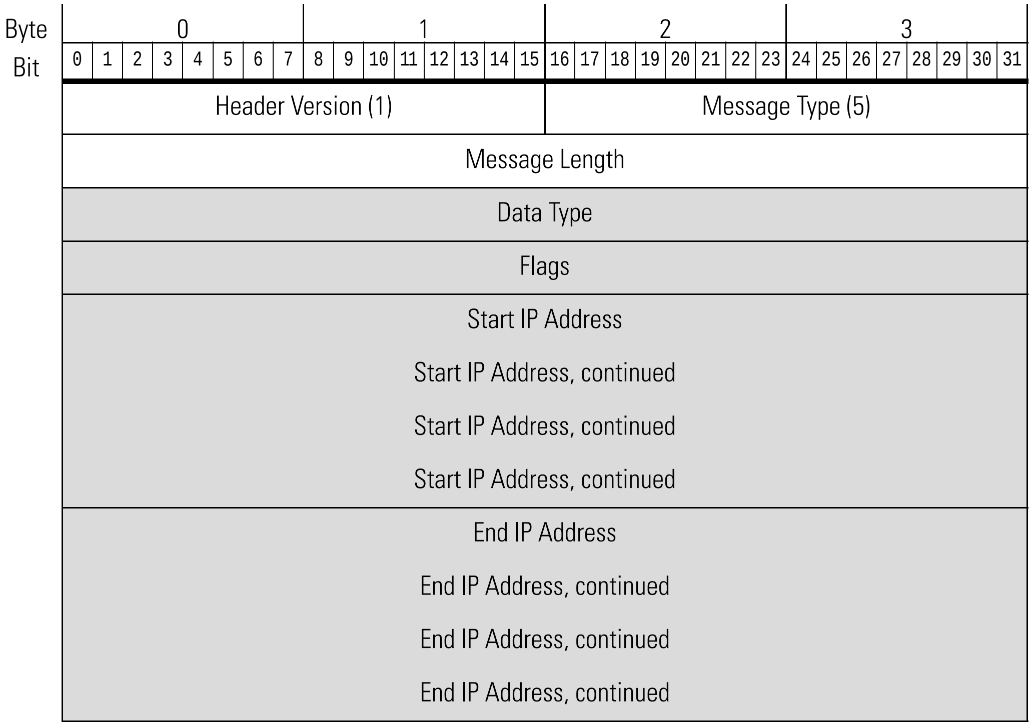
Note that it is mandatory for all data requests, including requests for host profile information, to first initialize the session by submitting an Event Stream Request message. To set up for streaming host data only, you can use any of the following request flag settings in your initial Event Stream Request message:

- set the bit for the appropriate version of metadata (this can be beneficial when streaming host data)
- set no request flags
- set bit 11 (to suppress any default event streaming if using legacy versions of eStreamer)

After the initial message, you then use a Host Request message (type 5) to specify the hosts.

IMPORTANT! For legacy eStreamer versions with default event streaming, if you want to stream only host profile data, you need to suppress the default event messages. First send the server an Event Stream Request message with bit 11 in the Request Flags field set to 1; then, send the Host Request message.

The graphic below shows the format for the Host Request message. The shaded fields are specific to the Host Request message format and are defined in the following table. The preceding three fields are the standard message header.



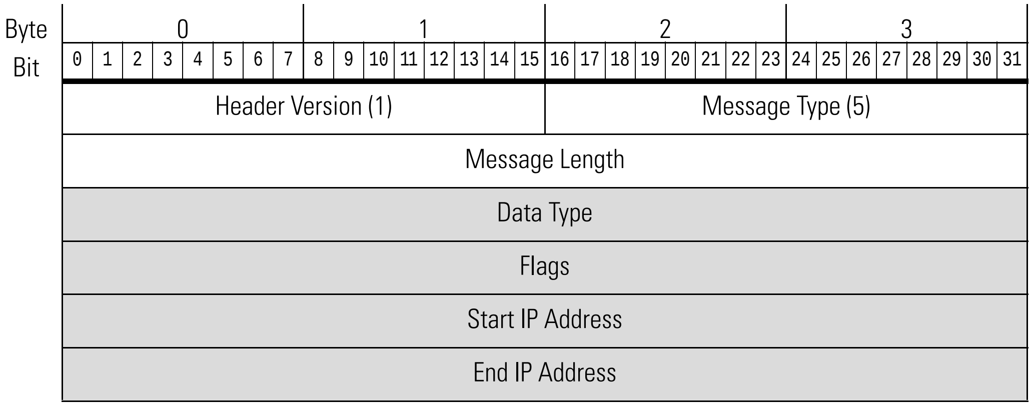
Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (5)	
	Message Length			
	Data Type			
	Flags			
	Start IP Address			
	Start IP Address, continued			
	Start IP Address, continued			
	Start IP Address, continued			
	End IP Address			
	End IP Address, continued			
	End IP Address, continued			
	End IP Address, continued			

The Host Request Message Fields table explains the message fields.

Host Request Message Fields

FIELD	DATA TYPE	DESCRIPTION
Data Type	uint32	Requests data for a single host or multiple hosts, using the following codes: • `0` — version 3.5 - 4.6 for a single host. • `1` — version 3.5 - 4.6 for multiple hosts (uses block 34). • `2` — version 4.7 - 4.8 for a single host (uses block 47). • `3` — version 4.7 - 4.8 for multiple hosts (uses block 47). • `4` — version 4.9 - 4.10 for a single host (uses block 92, see Full Host Profile Data Block 4.8 on page 656). • `5` — version 4.9 - 4.10 for multiple hosts (uses block 92, see Full Host Profile Data Block 4.8 on page 656). • `6` — version 5.0+ data for a single host (uses block 111, see Full Host Profile Data Block 5.3+ on page 388). • `7` — version 5.0+ data for multiple hosts (uses block 111, see Full Host Profile Data Block 5.3+ on page 388).
Flags	32-bit field	• `0x00000001` — Causes the Notes field of the host profile to be populated (with user-defined information about the host stored in the Sourcefire 3D System). • `0x00000002` — Causes the Banner field of the service block to be populated (with the first 256 bytes of the first packet detected for the service). Banners are disabled by default and available only if configured.
Start IP Address	uint8[16]	IP address of the host whose data should be returned (if request is for a single host), or the starting address in an IP address range (if request is for multiple hosts). Can be either an IPv4 or IPv6 address.
End IP Address	uint8[16]	Ending address in an IP address range (if request is for multiple hosts), or the Start IP Address value (if request is for single host). Can be either an IPv4 or IPv6 address.

The graphic below shows the format for the legacy Host Request message. eStreamer will still respond to this request. The only difference from the current request is the smaller IPv4 address fields. The shaded fields are specific to the Host Request message format and are defined in the following table. The preceding three fields are the standard message header.



The Host Request Message Fields table explains the message fields.

Host Request Message Fields

FIELD	DATA TYPE	DESCRIPTION
Data Type	uint32	Requests data for a single host or multiple hosts, using the following codes: • 0 — version 3.5 - 4.6 for a single host. • 1 — version 3.5 - 4.6 for multiple hosts (uses block 34). • 2 — version 4.7 - 4.8 for a single host (uses block 47). • 3 — version 4.7 - 4.8 for multiple hosts (uses block 47). • 4 — version 4.9 - 4.10 for a single host (uses block 92, see Full Host Profile Data Block 4.8 on page 656). • 5 — version 4.9 - 4.10 for multiple hosts (uses block 92, see Full Host Profile Data Block 4.8 on page 656). • 6 — version 5.0+ data for a single host (uses block 111, see Full Host Profile Data Block 5.3+ on page 388). • 7 — version 5.0+ data for multiple hosts (uses block 111, see Full Host Profile Data Block 5.3+ on page 388).

Host Request Message Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Flags	32-bit field	• `0x00000001` — Causes the Notes field of the host profile to be populated (with user-defined information about the host stored in the Sourcefire 3D System). • `0x00000002` — Causes the Banner field of the service block to be populated (with the first 256 bytes of the first packet detected for the service). Banners are disabled by default and available only if configured.
Start IP Address	uint8[4]	IP address of the host whose data should be returned (if request is for a single host), or the starting address in an IP address range (if request is for multiple hosts). Specify the address in IP address octets.
End IP Address	uint8[4]	Ending address in an IP address range (if request is for multiple hosts), or the Start IP Address value (if request is for single host).

Host Data and Multiple Host Data Message Format

eStreamer responds to host requests by sending host data messages, each with a full host profile data block. eStreamer sends one host data message for each host specified in the request. eStreamer uses the type 6 message to respond to requests for a single host profile, and uses the type 7 message to respond to requests for multiple hosts. The formats of the type 6 and type 7 messages are identical, only the message type is different.

Host data messages do not have a record type field. The structure of the message is communicated by the message type and the data block type of the full host profile included in the message. Full host profile data blocks are in the series a group of blocks.

The graphic below shows the format of the host data message and the table that follows defines the shaded fields:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (6 | 7)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Full Host Profile Data Block Type
See the Host Discovery and Connection Data Block Types table on page 225</td></tr>
<tr><td></td><td colspan="32">Length</td></tr>
<tr><td></td><td colspan="32">Full Host Profile Data Block</td></tr>
</table>

The fields specific to the Host Request message are:

FIELD	DATA TYPE	DESCRIPTION
Full Host Profile Data Block Type	uint32	Specifies the block type for the full host profile data included in the message. See the Host Discovery and Connection Data Block Types table on page 225.
Length	uint32	Length of the full host profile data in the message.
Full Host Profile Data Block	variable	The host data. For links to the definitions of current full host profile data blocks, see the Host Discovery and Connection Data Block Types table on page 225.

Streaming Information Message Format

When the eStreamer service receives a request for an extended request, it sends the client the Streaming Information message described below. This message advertises the server’s list of available services. Currently, the only relevant option is the eStreamer service (6667), although the message can list other services, which should be ignored. Each advertised service is represented by a Streaming Service Request structure described in the Streaming Service Request Structure table on page 54.

The graphic below illustrates the format for the Streaming Information message. The shaded field is specific to this message type. The preceding three fields are the standard message header.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (2051)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Service...
See Streaming Service Request Structure on page 54</td></tr>
</table>

The fields of the Streaming Information message are:

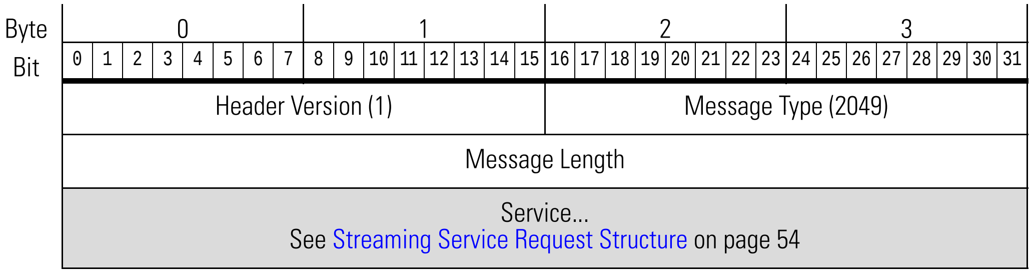
Streaming Information Message Fields

FIELD	DATA TYPE	DESCRIPTION
Header Version	uint16	Set to 1.
Message Type	uint16	eStreamer message type. Set to 2051 for Streaming Request messages.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the Header Version, Message Type, and Message Length fields.
Service[]	array	List of available services. See Streaming Service Request Structure on page 54.

Streaming Request Message Format

The client uses the Streaming Request message to specify to eStreamer the service in the Streaming Information message that it wants to use, followed by a set of requests for event types and versions to be streamed. The graphic below shows the message structure and the following table defines the fields. The requested service is represented by a Streaming Service Request structure described in the Streaming Service Request Structure table on page 54.

The graphic below illustrates the format for the Streaming Information message. The shaded field is specific to this message type. The preceding three fields are the standard message header.



The fields of the Streaming Request message are:

Streaming Request Message Fields

FIELD	DATA TYPE	DESCRIPTION
Header Version	uint16	Set to 1.
Message Type	uint16	eStreamer message type. Set to 2049 for Streaming Request messages.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the Header Version, Message Type, and Message Length fields.
Service[]	array	List of requested service structures. See Streaming Service Request Structure on page 54.

Streaming Service Request Structure

The eStreamer service sends one Streaming Service Request data structure in the Streaming Information message for each service it advertises. The eStreamer service does not use the last field of the Streaming Service Request, which provides for a list of event types to be included.

The client processes the Streaming Service Request structure from eStreamer and uses the same structure in the response it returns to the server. In the Streaming Service Request that the client sends to the server, it includes, first, a request for the service advertised by eStreamer, and, second, a list of Streaming Event Type structures, which specify the requested event types the client wants to receive.

Each Streaming Event Type structure contains two fields to specify the event type and version for each requested event type. For information on the Streaming Event Type structure, see Streaming Event Type Structure on page 57.

The graphic below shows the fields of the Streaming Service Request structure. The table that follows defines the fields.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Type</td></tr>
<tr><td></td><td colspan="32">Length</td></tr>
<tr><td></td><td colspan="32">Flags</td></tr>
<tr><td></td><td colspan="32">Initial Timestamp</td></tr>
<tr><td></td><td colspan="32">Streaming Event Type...
See Streaming Event Type Structure on page 57</td></tr>
</table>

The fields of the Streaming Service Request structure are:

Streaming Service Request Fields

FIELD	DATA TYPE	DESCRIPTION
Type	uint32	Service ID. In eStreamer server messages, this advertises an available service. In client messages, it specifies a requested service. Current valid options: • 6667 (for eStreamer service)
Length	uint32	Service request length. Describes the length of the fields following Length. Note that Length must include all the Streaming Event Type records in the message, plus the terminating one.
Flags	uint32	In eStreamer's Streaming Information messages: Always zero In client's Streaming Request message: replicates the flag settings in the original Event Stream Request message.

Streaming Service Request Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Initial Timestamp	uint32	In eStreamer's Streaming Information messages: Always zero In client's Streaming Request message: replicates the timestamp in the original Event Stream Request message.
Streaming Event Type	array	In eStreamer's Streaming Information message: • Reserved for future use. Has 0 length. In client's Streaming Request message: • One Streaming Event Type entry for each requested event type. See Streaming Event Type Structure below. • Terminate the request list with a zero Event Type entry, with both Event Type and Version set to 0. See Streaming Event Type Structure on page 57.

Streaming Event Type Structure

eStreamer clients use the Streaming Event Type structure to specify an event's version and version. Each event version/type combination is a request for an event stream.

Lists of Streaming Event Type structures must be terminated with a structure with all fields set to zero. That is:

```
Event Version = 0
Event Type = 0
```

The following diagram illustrates the format for the Streaming Event Type structure.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Event Version																Event Type															

The fields of the Streaming Event Type structure are:

Streaming Event Type Fields

FIELD	DATA TYPE	DESCRIPTION
Event Version	uint16	Version number of event type. For list of versions supported for each event type, see Event Types and Versions for Extended Request table below.
Event Type	uint16	Code for requested event type. For the current list of valid event types and version codes, see the Event Types and Versions for Extended Request table below. List of event types should be terminated with a zero event type and zero event version.

The Event Types and Versions for Extended Request table below lists the event types and versions that clients can specify in extended requests. The table indicates the Defense Center (DC) software versions that correspond to each event type version. For example, to request the correlation events that were supported by the Defense Center in version 4.8.0.2 - 4.9.1, you should request Event Type 31, Version 5. If an event was recorded with a different event type, it will be upgraded or downgraded to match the format of the requested event type.

Event Types and Versions for Extended Request

TO REQUEST...	USE THIS EVENT VERSION NUMBER..	AND THIS EVENT CODE
intrusion events	1 — for DC 4.8.x and earlier 2 — for DC 4.9 - 4.10.x 3 — for DC 5.0 - 5.1 4 — for DC 5.1.1.x 5 — for DC 5.2.x 6 — for DC 5.3+	12
metadata	1 — for DC 3.2 - 4.5.x 2 — for DC 4.6.0.x 3 — for DC 4.6.1 - 4.6.x 4 — for DC 4.7+	21

Event Types and Versions for Extended Request (Continued)

TO REQUEST...	USE THIS EVENT VERSION NUMBER...	AND THIS EVENT CODE
correlation and compliance white list events	1 — for DC 3.2 and earlier 2 — for DC 4.0 - 4.4.x 3 — for DC 4.5 - 4.6.1 4 — for DC 4.7 - 4.8.0.1 5 — for DC 4.8.0.2 - 4.9.1.x 6 — for DC 4.10.0 - 4.10.x 7 — for DC 5.0 - 5.0.2 8 — for DC 5.1+	31
discovery events	1 — for DC 3.2 and earlier 2 — for DC 3.0 - 3.4.x 3 — for DC 3.5 - 4.6.x 4 — for DC 4.7 - 4.8.x 5 — for DC 4.9.0.x 6 — for DC 4.9.1 - 4.9.x.x 7 — for DC 4.10.0 - 4.10.x 8 — for DC 5.0.x 9 — for DC 5.1.x 10 — for DC 5.2+	61
connection events	1 — for DC 4.0 - 4.1 3 — for DC 4.5 - 4.6.1 4 — for DC 4.7 - 4.9.0.x 5 — for DC 4.9.1 - 4.10.x 6 — for DC 5.0.x 7 — for DC 5.1.0.x 8 — for DC 5.1.1.x 9 — for DC 5.2+	71
user events	1 — for DC 4.7 - 4.10.x 2 — for DC 5.0.x 3 — for DC 5.1-5.1.x 4 — for DC 5.2+	91
malware events	1 — for DC 5.1.0.x 2 — for DC 5.1.1.x 3 — for DC 5.2.x 4 — for DC 5.3+	101
file events	1 — for DC 5.1.1 - 5.1.x 2 — for DC 5.2.x 3 — for DC 5.3+	111
impact correlation events	1 — for DC5.2.x and earlier 2 — for DC 5.3+	131
terminating event type in a list	0	0

Sample Extended Request Messages

Streaming Information Message

In the sample below, the server advertises two services, the first type 6667 (eStreamer) and the second type 5000. In Streaming Information messages from the server, the flags field and initial timestamp fields are zero, and the message specifies no event types.

Header Version:	`1`	/*always 1*/
Message Type:	`2051`	/*streaming info msg*/
Message Length	`32`	/*bytes of msg content*/
Service[1].Type	`6667`	/*eStreamer service ID*/
Service[1].Length	`8`	
Service[1].Flags	`0`	/*no flags from server*/
Service[1].Initial Timestamp	`0`	/*always 0*/
Service[2].Type	`5000`	/*service-2 ID*/
Service[2].Length	`8`	
Service[2].Flags	`0`	/*no flags from server*/
Service[2].Initial Timestamp	`0`	/*always 0*/
Header Version:	`1`	/*always 1*/
Message Type:	`2051`	/*streaming info msg*/

Streaming Request Message

Below is a Streaming Request message where the client requests service type 6667 (eStreamer) and specifies two event types: version 6 of connection events (event type 71) and version 4 of metadata (event type 21).

Header Version:	`1`	/*always 1*/
Message Type:	`2049`	/*stream request msg*/
Message Length	`28`	/*payload bytes*/
Service[1].Type	`6667`	/*eStreamer service ID*/
Service[1].Length	`20`	
Service[1].Flags	`30`	/*original flags value*/
Service[1].Initial Timestamp	`0`	/*original timestamp*/
Service[1].Event[1].Version	`6`	/*version 6*/
Service[1].Event[1].Type	`71`	/*connection events*/
Service[1].Event[2].Version	`4`	/* version 4*/
Service[1].Event[2].Type	`21`	/*metadata*/
Service[1].Event[3].Version	`0`	/*terminate event list*/
Service[1].Event[3].Type	`0`	/*terminate event list*/

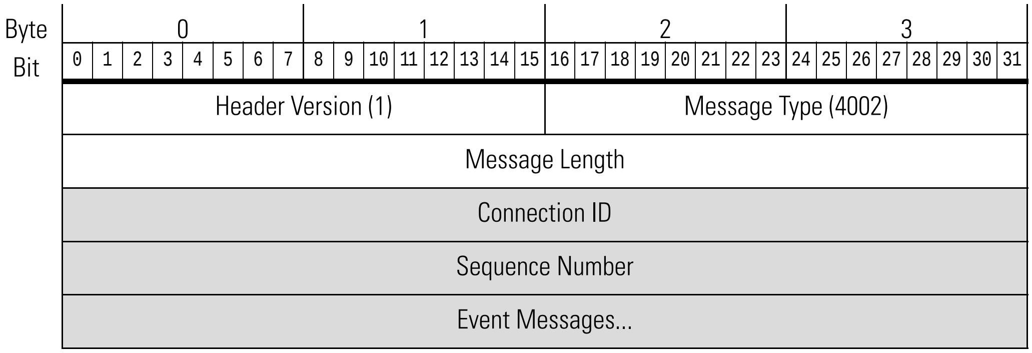
Message Bundle Format

The eStreamer server sends messages in a bundle format when the client submits an extended request.

The client responds with a NULL message to acknowledge receipt of an entire bundle. The client should not acknowledge receipt of individual messages in a bundle.

Message bundles have a message type of 4002.

The graphic below shows the structure of a message bundle. The shaded fields are specific to the bundle message type. The following table describes the content of the fields and data structures.



The fields of a message bundle message are:

Message Bundle Message Fields

FIELD	DATA TYPE	DESCRIPTION
Header Version	uint16	Always 1.
Message Type	uint16	Always 4002.
Message Length	uint32	Length of the content of the message after the message header. Does not include the bytes in the bundle's Header Version, Message Type, and Message Length fields. As the client loads a message from the bundle, it can subtract the message's total length (including header) from the length in this field. As long as the remainder is positive, there are more messages to process.
Connection ID	uint32	A unique identifier for the connection with the server.
Sequence Number	uint32	Starts at 1 and increments by one for each bundle sent by the eStreamer server.
Event Messages []	array	The events streamed by the server in the bundle. Each message has a full set of headers, including message version number (1), archive timestamp if requested, and so forth.

Understanding Metadata

The eStreamer server can provide metadata along with requested event records. To receive metadata, you must explicitly request it. See the Request Flags table on page 31 for information on how to request a given version of metadata. The metadata provides context information for codes and numeric identifiers in the event records. For example, an intrusion event contains only the internal identifier of the detecting device, and the metadata provides the device's name.

Metadata Transmission

If the request message specifies metadata, eStreamer sends the relevant metadata record before it sends any related event records.

eStreamer keeps track of the metadata it has sent to the client and does not resend the same metadata record. The client should cache each received metadata record. eStreamer does not keep a history of metadata transmissions from one session to the next, so when a new session starts and a request message specifies metadata, eStreamer restarts metadata streaming from scratch.

CHAPTER 3

UNDERSTANDING INTRUSION AND CORRELATION DATA STRUCTURES

The eStreamer service transmits a number of data record types to deliver requested events and metadata to the client. This chapter describes the structures of data records for the following types of event data:

- intrusion events data and event extra data generated by managed devices
- correlation (compliance) events generated by the Defense Center
- metadata records

The following sections in this chapter define the event message structures:

- Intrusion Event and Metadata Record Types below on page 106

For a general overview eStreamer's message format for transmitting data records, see Event Data Message Format on page 37.

Intrusion Event and Metadata Record Types

The Intrusion Event and General Metadata Record Types table below lists all currently supported record types for intrusion events, intrusion event extra data, and metadata messages. The data for these record types is in fixed-length fields. By contrast, correlation event records contain one or more levels of nested data blocks with variable lengths. The table below provides a link to the chapter subsection that defines the associated data record structure.

For some record types, eStreamer supports more than one version. The table indicates the status of each version (current or legacy). A current record is the latest version. A legacy record has been superseded by a later version but can still be requested from eStreamer.

Intrusion Event and General Metadata Record Types

Record Type	Block Type	Series	Description	Record Status	Data Format Described In...
2	N/A	N/A	Packet Data (Version 4.8.0.2+)	Current	Packet Record 4.8.0.2+ on page 67
4	N/A	N/A	Priority Metadata	Current	Priority Record on page 69
9	20	1	Intrusion Impact Alert	Current	Intrusion Impact Alert Data on page 77
62	N/A	N/A	User Metadata	Current	User Record on page 81
66	N/A	N/A	Rule Message Metadata (Version 4.6.1+)	Current	Rule Message Record for 4.6.1+ on page 82
67	N/A	N/A	Classification Metadata (Version 4.6.1+)	Current	Classification Record for 4.6.1+ on page 83
69	N/A	N/A	Correlation Policy Metadata (Version 4.6.1+)	Current	Correlation Policy Record on page 85
70	N/A	N/A	Correlation Rule Metadata (Version 4.6.1+)	Current	Correlation Rule Record on page 87
104	N/A	N/A	Intrusion Event (IPv4) Record 4.9 - 4.10.x	Legacy	Intrusion Event (IPv4) Record for 4.9 - 4.10.x on page 458
105	N/A	N/A	Intrusion Event (IPv6) Record 4.9-4.10.x	Legacy	Intrusion Event (IPv6) Record for 4.10.2.3 on page 462
110	4	2	Intrusion Event Extra Data (Version 4.10.0+)	Current	Intrusion Event Extra Data Record on page 89
111	5	2	Intrusion Event Extra Data Metadata (Version 4.10.0+)	Current	Intrusion Event Extra Data Metadata on page 91
112	128	1	Correlation Event for 5.1+	Current	Correlation Event for 5.1+ on page 106
115	14	2	Security Zone Name Metadata	Current	Security Zone Name Record on page 93

Intrusion Event and General Metadata Record Types (Continued)

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
116	14	2	Interface Name Metadata	Current	Interface Name Record on page 94
117	14	2	Access Control Policy Name Metadata	Current	Access Control Policy Name Record on page 96
118	15	2	Intrusion Policy Name Metadata	Current	Intrusion Policy Name Record on page 190
119	15	2	Access Control Rule ID Metadata	Current	Access Control Rule ID Record Metadata on page 97
120	N/A	N/A	Access Control Rule Action Metadata	Current	Access Control Rule Action Record Metadata on page 191
121	N/A	N/A	URL Category Metadata	Current	URL Category Record Metadata on page 192
122	21	2	URL Reputation Metadata	Current	URL Reputation Record Metadata on page 193
123	N/A	N/A	Managed Device Metadata	Current	Managed Device Record Metadata on page 99
125	33	2	Malware Event Record (Version 5.1.1+)	Current	Malware Event Record 5.1.1+ on page 100
127	14	2	Sourcefire Cloud Name Metadata (Version 5.1+)	Current	Sourcefire Cloud Name Metadata on page 101
128	N/A	N/A	Malware Event Type Metadata (Version 5.1+)	Current	Malware Event Type Metadata on page 102
129	N/A	N/A	Malware Event Subtype Metadata (Version 5.1+)	Current	Malware Event Subtype Metadata on page 103
130	N/A	N/A	FireAMP Detector Type Metadata (Version 5.1+)	Current	FireAMP Detector Type Metadata on page 104
131	N/A	N/A	FireAMP File Type Metadata (Version 5.1+)	Current	FireAMP File Type Metadata on page 105
161	39	2	IOC Name Data Block for 5.3+	Current	IOC Name Data Block for 5.3+ on page 160

Intrusion Event and General Metadata Record Types (Continued)

Record Type	Block Type	Series	Description	Record Status	Data Format Described in...
207	N/A	N/A	Intrusion Event (IPv4) Record 5.0.x - 5.1	Legacy	Intrusion Event (IPv4) Record 5.0.x - 5.1 on page 466
208	N/A	N/A	Intrusion Event (IPv6) Record 5.0.x - 5.1	Legacy	Intrusion Event (IPv6) Record 5.0.x - 5.1 on page 472
260	19	2	ICMP Type Data Data Block	Current	ICMP Type Data Block on page 128
270	20	2	ICMP Code Data Block	Current	ICMP Code Data Block on page 129
400	34	2	Intrusion Event Record 5.2+	Current	Intrusion Event Record 5.3+ on page 70
500	32	2	File Event (Version 5.2+)	Legacy	File Event for 5.2.x on page 623
500	38	2	File Event (Version 5.3+)	Current	File Event for 5.3+ on page 133
502	33	2	Malware Event (Version 5.2x)	Legacy	Malware Event Data Block 5.2.x on page 505
502	35	2	Malware Event (Version 5.3+)	Current	Malware Event Data Block 5.3+ on page 140
511	26	2	File Event SHA Hash (Version 5.1.1+)	Current	Rule Documentation Data Block for 5.2+ on page 151
520	28	2	Geolocation Data Block for 5.2+	Current	Geolocation Data Block for 5.2+ on page 156
N/A	150	1	IOC State Data Block for 5.3+	Current	IOC State Data Block for 5.3+ on page 158

Packet Record 4.8.0.2+

The eStreamer service transmits the packet data associated with an event in a Packet record, the format of which is shown below. Packet data is sent when the Packet flag—bit 0 in the Request Flags field of a request message—is set. See

Request Flags on page 30. If you enable bit 23, an extended event header is included in the record. Note that the Record Type field, which appears after the Message Length field, has a value of 2, indicating a packet record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (2)			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
	Device ID			
	Event ID			
	Event Second			
	Packet Second			
	Packet Microsecond			
	Link Type			
	Packet Length			
	Packet Data...			

The Packet Record Fields table describes the fields in the Packet record.

Packet Record Fields

FIELD	DATA TYPE	DESCRIPTION
Device ID	uint32	The device identification number. You can obtain device names that correlate to them by requesting Version 3 or 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Event ID	uint32	The event identification number.
Event Second	uint32	The second (from 01/01/1970) that the event occurred.

Packet Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Packet Second	uint32	The second (from 01/01/1970) that the packet was captured.
Packet Microsecond	uint32	Microsecond (one millionth of a second) increment that the packet was captured.
Link Type	uint32	Link layer type. Currently, the value will always be 1 (signifying the Ethernet layer).
Packet Length	uint32	Number of bytes included in the packet data.
Packet Data	variable	Actual captured packet data (header and payload).

Priority Record

The eStreamer service transmits the priority associated with an event in a Priority record, the format of which is shown below. (Priority information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 4, indicating a Priority record.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (4)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">Priority ID</td></tr>
<tr><td></td><td colspan="16">Name Length</td><td colspan="16">Priority Name...</td></tr>
</table>

The Priority Record Fields table describes each priority-specific field.

Priority Record Fields

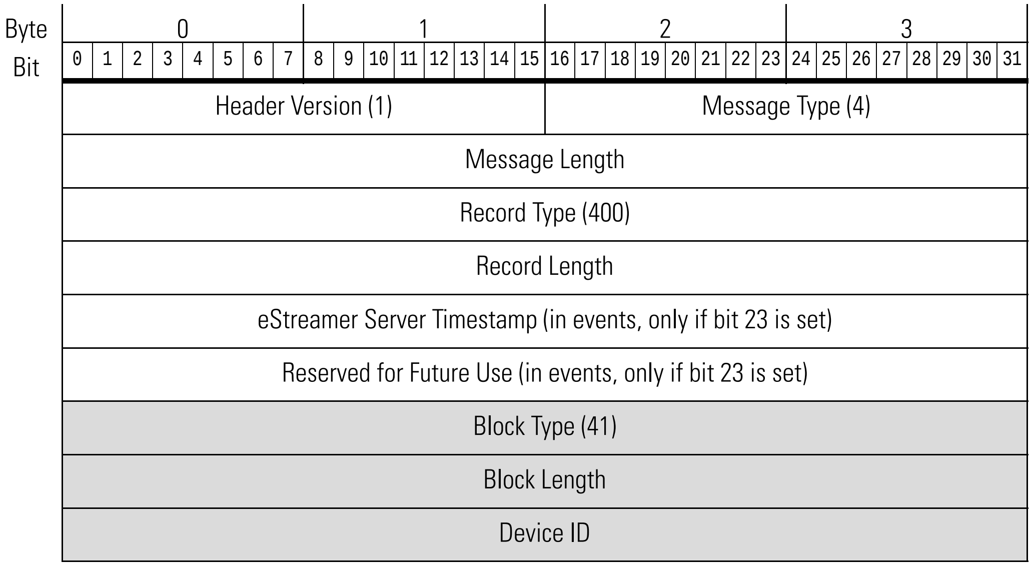
Field	Data Type	Description
Priority ID	uint32	Indicates the priority identification number.
Name Length	uint16	Number of bytes included in the priority name.
Priority Name	variable	Priority name that corresponds with the priority ID (1 — high, 2 — medium, 3 — low).

Intrusion Event Record 5.3+

The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 41.

You can request 5.3+ intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 6 in the Stream Request message (see Submitting Extended Requests on page 20 for information about submitting extended requests).

For version 5.3+ intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.



<table>
<tr><th>Byte</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (400)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">eStreamer Server Timestamp (in events, only if bit 23 is set)</td></tr>
<tr><td></td><td colspan="32">Reserved for Future Use (in events, only if bit 23 is set)</td></tr>
<tr><td></td><td colspan="32">Block Type (41)</td></tr>
<tr><td></td><td colspan="32">Block Length</td></tr>
<tr><td></td><td colspan="32">Device ID</td></tr>
</table>

<table>
<tr><td colspan="4">Event ID</td></tr>
<tr><td colspan="4">Event Second</td></tr>
<tr><td colspan="4">Event Microsecond</td></tr>
<tr><td colspan="4">Rule ID (Signature ID)</td></tr>
<tr><td colspan="4">Generator ID</td></tr>
<tr><td colspan="4">Rule Revision</td></tr>
<tr><td colspan="4">Classification ID</td></tr>
<tr><td colspan="4">Priority ID</td></tr>
<tr><td colspan="4">Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued</td></tr>
<tr><td colspan="4">Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued</td></tr>
<tr><td colspan="2">Source Port or ICMP Type</td><td colspan="2">Destination Port or ICMP Code</td></tr>
<tr><td>IP Protocol ID</td><td>Impact Flags</td><td>Impact</td><td>Blocked</td></tr>
<tr><td colspan="4">MPLS Label</td></tr>
<tr><td colspan="2">VLAN ID</td><td colspan="2">Pad</td></tr>
<tr><td colspan="4">Policy UUID
Policy UUID, continued
Policy UUID, continued
Policy UUID, continued</td></tr>
<tr><td colspan="4">User ID</td></tr>
<tr><td colspan="4">Web Application ID</td></tr>
<tr><td colspan="4">Client Application ID</td></tr>
<tr><td colspan="4">Application Protocol ID</td></tr>
</table>

<table>
<tr><td colspan="2">Access Control Rule ID</td></tr>
<tr><td colspan="2">Access Control Policy UUID
Access Control Policy UUID, continued
Access Control Policy UUID, continued
Access Control Policy UUID, continued</td></tr>
<tr><td colspan="2">Interface Ingress UUID
Interface Ingress UUID, continued
Interface Ingress UUID, continued
Interface Ingress UUID, continued</td></tr>
<tr><td colspan="2">Interface Egress UUID
Interface Egress UUID, continued
Interface Egress UUID, continued
Interface Egress UUID, continued</td></tr>
<tr><td colspan="2">Security Zone Ingress UUID
Security Zone Ingress UUID, continued
Security Zone Ingress UUID, continued
Security Zone Ingress UUID, continued</td></tr>
<tr><td colspan="2">Security Zone Egress UUID
Security Zone Egress UUID, continued
Security Zone Egress UUID, continued
Security Zone Egress UUID, continued</td></tr>
<tr><td colspan="2">Connection Timestamp</td></tr>
<tr><td>Connection Instance ID</td><td>Connection Counter</td></tr>
<tr><td>Source Country</td><td>Destination Country</td></tr>
<tr><td>IOC Number</td><td></td></tr>
</table>

The Intrusion Event Record 5.3+ Fields table describes each intrusion event record data field.

Intrusion Event Record 5.3+ Fields

FIELD	DATA TYPE	DESCRIPTION
Block Type	unint32	Initiates an Intrusion Event data block. This value is always 34.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event’s detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event’s detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.

Intrusion Event Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event Record 5.3+ Fields (Continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — Gray (unknown impact)

Intrusion Event Record 5.3+ Fields (Continued)

Field	Data Type	Description
Blocked	uint8	Value indicating whether the event was blocked. • `0` — not blocked • `1` — blocked • `2` — would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.

Intrusion Event Record 5.3+ Fields (Continued)

Field	Data Type	Description
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
IOC Number	uint16	ID Number of the compromise associated with this event.

Intrusion Impact Alert Data

The Intrusion Impact Alert event contains information about impact events. It is transmitted when an intrusion event is compared to the system network map data and the impact is determined. It uses the standard record header with a record type of 9, followed by an Intrusion Impact Alert data block with a series 1 data block type of 20 in the series 1 group of blocks. (The Impact Alert data block is a type of series 1 data block. For more information about series 1 data blocks, see Understanding Discovery (Series 1) Blocks on page 224.)

You can request that eStreamer only transmit intrusion impact events by setting bit 5 in the Flags field of the request message. See Event Stream Request Message Format on page 28 for more information about request messages. Version 1 of these alerts only handles IPv4. Version 2, introduced in 5.3, handles IPv6 events in addition to IPv4.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (9)			
	Record Length			
	Intrusion Impact Alert Block Type (20)			
	Intrusion Impact Alert Block Length			
	Event ID			
	Device ID			
	Event Second			
	Impact			
	Source IP Address			
	Destination IP Address			
Impact Description	String Block Type (0)			
	String Block Length			
	Description...			

The Impact Event Data Fields table describes each data field in an impact event.

Impact Event Data Fields

FIELD	DATA TYPE	DESCRIPTION
Intrusion Impact Alert Block Type	uint32	Indicates that an intrusion impact alert data block follows. This field will always have a value of 20. See Intrusion Event and Metadata Record Types on page 64.
Intrusion Impact Alert Block Length	uint32	Indicates the length of the intrusion impact alert data block, including all data that follows and 8 bytes for the intrusion impact alert block type and length.
Event ID	uint32	Indicates the event identification number.
Device ID	uint32	Indicates the managed device identification number.
Event Second	uint32	Indicates the second (from 01/01/1970) that the event was detected.

Impact Event Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Impact	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Source IP Address	uint8[4]	IP address of the host associated with the impact event, in IP address octets.
Destination IP Address	uint8[4]	IP address of the destination IP address associated with the impact event (if applicable), in IP address octets. This value is 0 if there is no destination IP address.

Impact Event Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a string data block that contains the impact name. This value is always set to 0. For more information about string blocks, see String Data Block on page 237.
String Block Length	uint32	Number of bytes in the event description string block. This includes the four bytes for the string block type, the four bytes for the string block length, and the number of bytes in the description.
Description	string	Description of the impact event.

User Record

When you request metadata, you can retrieve information about the users referenced in events generated by components in your Sourcefire 3D System. The eStreamer service transmits metadata containing user information for an event within a User record, the format of which is shown below. The user metadata record can be used to determine a user name associated with an event by correlating the metadata with the user ID value from a User Vulnerability Change Data Block, User Host Deletion Data Block, User Service Deletion Data Block, User Criticality Change Blocks, Attribute Definition Data Block, User Attribute Value Data Block, or Scan Result Data Block. (User information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 62, indicating a User record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (62)			
	Record Length			
	User ID			
	Name Length			
	Name...			

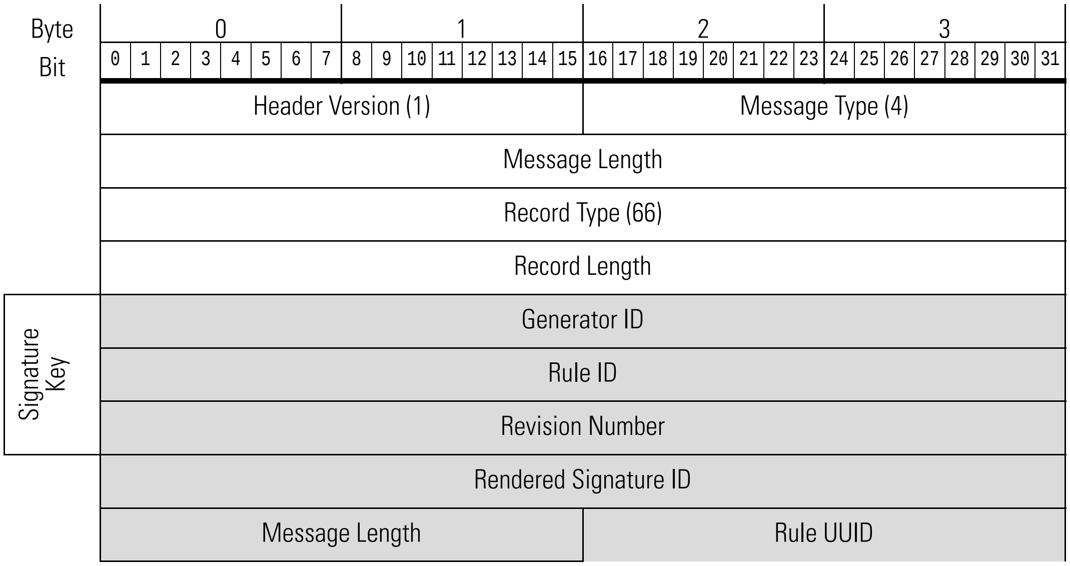
The User Record Fields table describes the fields in the User record.

User Record Fields

FIELD	DATA TYPE	DESCRIPTION
User ID	uint32	The user ID number.
Name Length	uint32	The number of bytes included in the user name.
Name	string	The name of the user.

Rule Message Record for 4.6.1+

Rule message information for an event is transmitted within a Rule Message record, the format of which is shown below. The eStreamer service transmits the Rule Message record for 4.6.1+ when you request Version 2 or Version 3 metadata. The Rule Message record for 4.6.1+ contains the same fields as the Rule Message record for 4.6 and lower but also has new UUID and Revision UUID fields. (Version 2, Version 3, or Version 4 metadata information is sent when the appropriate metadata flag—bit 14 for Version 2, bit 15 for Version 3, or bit 20 for Version 4 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 66, indicating a Rule Message Version 2 record.



Rule UUID	Rule UUID cont.	
	Rule UUID cont.	
	Rule UUID cont.	
	Rule UUID cont.	Rule Revision UUID
Rule Revision UUID	Rule Revision UUID cont.	
	Rule Revision UUID cont.	
	Rule Revision UUID cont.	
	Rule Revision UUID cont.	Message...

The Rule Message Record Fields table describes each rule-specific field.

Rule Message Record Fields

FIELD	DATA TYPE	DESCRIPTION
Generator ID	uint32	The generator identification number.
Rule ID	uint32	The rule identification number for the local computer.
Rule Revision	uint32	The rule revision number. This is currently set to zero for all rule messages.
Rendered Signature ID	uint32	The rule identification number rendered to the Sourcefire 3D System interface.
Message Length	uint16	The number of bytes included in the rule text.
UUID	uint8[16]	A rule ID number that acts as a unique identifier for the rule.
Revision UUID	uint8[16]	A rule revision ID number that acts as a unique identifier for the revision.
Message	variable	Rule message that triggered the event.

Classification Record for 4.6.1+

The eStreamer service transmits the classification information for an event in a Classification record for 4.6.1+, the format of which is shown below. The Classification record for 4.6.1+ contains the same fields as the Classification record for 4.6 and lower but also has new UUID and Revision UUID fields. (Classification information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 67, indicating a Classification Version 2 record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (67)			
	Record Length			
	Classification ID			
	Name Length		Name...	
	Name, continued...			
	Description Length		Description...	
	Description, continued...			
Classification UUID	Classification UUID			
	Classification UUID, continued			
	Classification UUID, continued			
	Classification UUID, continued			
Classification Revision UUID	Classification Revision UUID			
	Classification Revision UUID, continued			
	Classification Revision UUID, continued			
	Classification Revision UUID, continued			

The Classification Record Fields table describes the fields in the Classification record.

Classification Record Fields

FIELD	DATA TYPE	DESCRIPTION
Classification ID	uint32	The classification ID number.
Name Length	uint16	The number of bytes included in the name.
Name	string	The classification name.
Description Length	uint16	The number of bytes included in the description.
Description	string	The classification description.
UUID	uint8[16]	A classification ID number that acts as a unique identifier for the classification.
Revision UUID	uint8[16]	A classification revision ID number that acts as a unique identifier for the classification revision.

Correlation Policy Record

The eStreamer service transmits metadata containing the correlation policy for a correlation event within a Correlation Policy record, the format of which is shown below. (Correlation policy information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 69, indicating a Correlation Policy record.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (69)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">Correlation Policy ID</td></tr>
</table>

	Name Length
	Name...
	Description Length
	Description...
Correlation Policy UUID	Correlation Policy UUID
	Correlation Policy UUID, continued
	Correlation Policy UUID, continued
	Correlation Policy UUID, continued
Correlation Policy Revision UUID	Correlation Policy Revision UUID
	Correlation Policy Revision UUID, continued
	Correlation Policy Revision UUID, continued
	Correlation Policy Revision UUID, continued

The Correlation Policy Record Fields table describes the fields in the Correlation Policy record.

Correlation Policy Record Fields

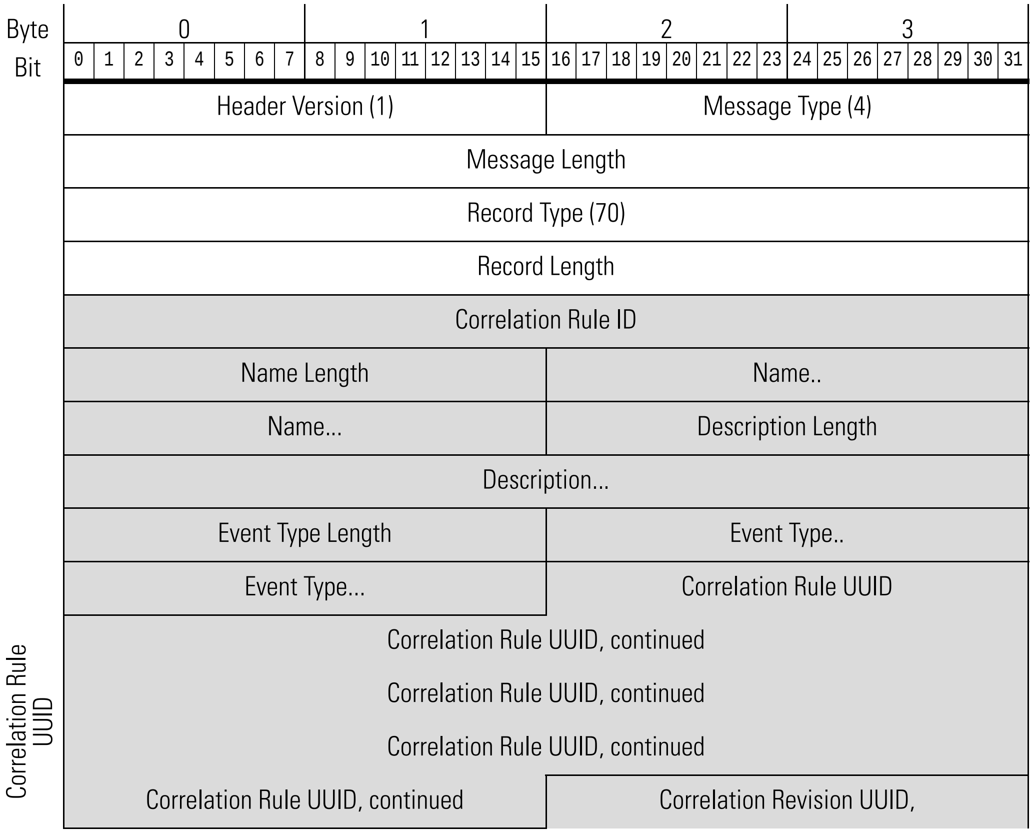
FIELD	DATA TYPE	DESCRIPTION
Correlation Policy ID	uint32	The correlation policy ID number.
Name Length	uint16	The number of bytes included in the correlation policy name.
Name	string	The name of the correlation policy that triggered the event.
Description Length	uint16	The number of bytes included in the correlation policy description.
Description	string	The description of the correlation policy that triggered the event.

Correlation Policy Record Fields (Continued)

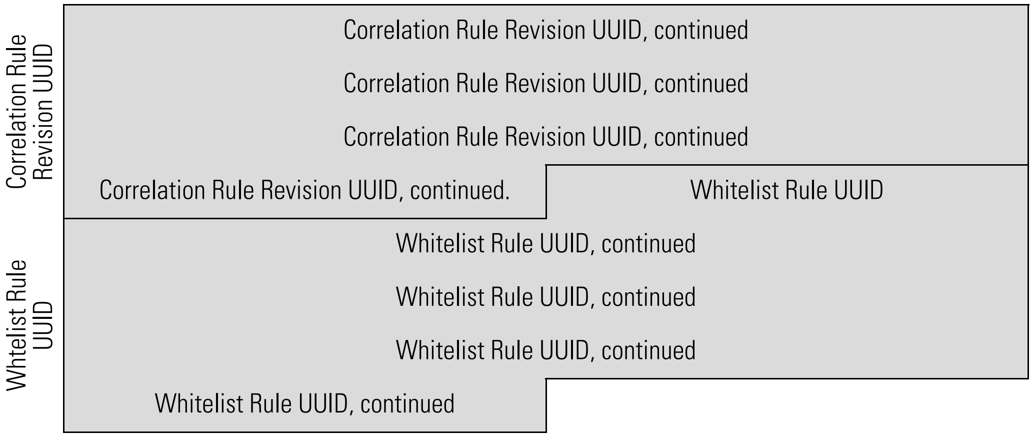
FIELD	DATA TYPE	DESCRIPTION
UUID	uint8[16]	A correlation policy ID number that acts as a unique identifier for the correlation policy.
Revision UUID	uint8[16]	A correlation policy revision ID number that acts as a unique identifier for the correlation policy.

Correlation Rule Record

The eStreamer service transmits metadata containing information on the correlation rule that triggered a correlation event within a Correlation Rule record, the format of which is shown below. (Correlation rule information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 70, indicating a Correlation Rule record.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (70)			
	Record Length			
	Correlation Rule ID			
	Name Length		Name..	
	Name...		Description Length	
	Description...			
	Event Type Length		Event Type..	
	Event Type...		Correlation Rule UUID	
Correlation Rule UUID	Correlation Rule UUID, continued			
	Correlation Rule UUID, continued			
	Correlation Rule UUID, continued			
	Correlation Rule UUID, continued		Correlation Revision UUID,	


Correlation Rule Revision UUID
Correlation Rule Revision UUID, continued
Correlation Rule Revision UUID, continued
Correlation Rule Revision UUID, continued
Correlation Rule Revision UUID, continued.
Whitelist Rule UUID
Whtelist Rule UUID
Whitelist Rule UUID, continued
Whitelist Rule UUID, continued
Whitelist Rule UUID, continued
Whitelist Rule UUID, continued

The Correlation Rule Record table describes the fields in the Correlation Rule record.

Correlation Rule Record Fields

FIELD	DATA TYPE	DESCRIPTION
Correlation Rule ID	uint32	The correlation rule ID number.
Name Length	uint16	The number of bytes included in the correlation rule name.
Name	string	The name of the correlation rule that triggered the event.
Description Length	uint16	The number of bytes included in the correlation rule description.
Description	string	The description of the correlation rule that triggered the event.
Event Type Length	uint16	The number of bytes included in the event type description.
Event Type	string	The description of the event that triggered the correlation rule.
UUID	uint8[16]	A correlation rule ID number that acts as a unique identifier for the correlation rule.

Correlation Rule Record Fields (Continued)

Field	Data Type	Description
Revision UUID	uint8[16]	A correlation rule revision ID number that acts as a unique identifier for the correlation rule revision.
Whitelist UUID	uint8[16]	A correlation ID number that acts as a unique identifier for the event sent as a result of a whitelist violation.

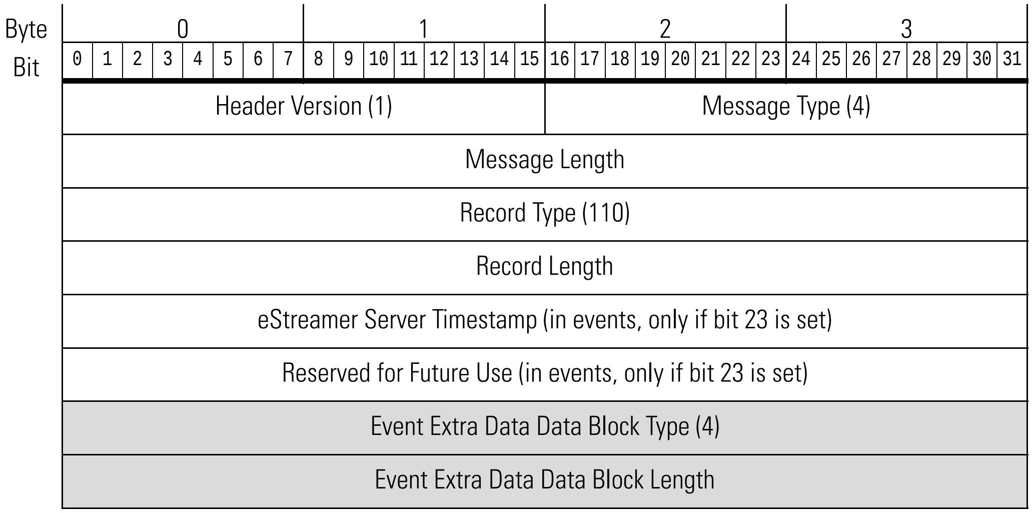
Intrusion Event Extra Data Record

The eStreamer service transmits the event extra data associated with an intrusion event in the Intrusion Event Extra Data record. The record type is always 110.

The event extra data appears in an encapsulated Event Extra Data data block, which always has a data block type value of 4. (The Event Extra Data data block is a series 2 data block. For more information about series 2 data blocks, see Understanding Series 2 Data Blocks on page 116.)

The supported types of extra data include IPv6 source and destination addresses, as well as the originating IP addresses (v4 or v6) of clients connecting to a web server through an HTTP proxy or load balancer. The graphic below shows the format of the Intrusion Event Extra Data record.

If bit 27 is set in the Request Flags field of the request message, you receive the event extra data for each intrusion event. If you set bit 20, you also receive the event extra data metadata described in Intrusion Event Extra Data Metadata on page 91. If you enable bit 23, eStreamer will include the extended event header. See Request Flags on page 30 for information on setting request flags.



Byte 0–1 (Bit 0–15)	Byte 2–3 (Bit 16–31)
Header Version (1)	Message Type (4)
Message Length	
Record Type (110)	
Record Length	
eStreamer Server Timestamp (in events, only if bit 23 is set)	
Reserved for Future Use (in events, only if bit 23 is set)	
Event Extra Data Data Block Type (4)	
Event Extra Data Data Block Length	

Device ID
Event ID
Event Second
Type
BLOB Block Type (1)
BLOB Length
Event Extra Data

Note that the Event Extra Data block structure includes a BLOB block type, which is one of several variable length data structures introduced in Version 4.10 of the Sourcefire 3D System.

The Intrusion Event Extra Data Data Block Fields table describes the fields in the Intrusion Event Extra Data record.

Intrusion Event Extra Data Data Block Fields

Field	Data Type	Description
Event Extra Data Data Block Type	uint32	Initiates an Event Extra Data data block. This value is always 4. The block type is a series 2 block; for information see Understanding Series 2 Data Blocks on page 116.
Event Extra Data Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Device ID	uint32	The managed device identification number.
Event ID	uint32	The event identification number.
Event Second	uint32	UNIX timestamp of the event (seconds since 01/01/1970).
Type	uint32	Identifier for the type of extra data; for example: • 1 — XFF client (IPv4) • 2 — XFF client (IPv6) • 9 — HTTP URI ...
BLOB Block Type	uint32	Initiates a BLOB data block containing extra data. This value is always 1. The block type is a series 2 block.

Intrusion Event Extra Data Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Length	uint32	Total number of bytes in the BLOB data block.
Extra Data	variable	The content of the extra data. The data type is indicated in the Type field.

Intrusion Event Extra Data Metadata

The eStreamer service transmits the event extra data metadata associated with intrusion event extra data records in the Intrusion Event Extra Data Metadata record. The record type is always 111.

The event extra data metadata appears in an encapsulated Event Extra Data Metadata data block, which always has a data block type value of 5. The Event Extra Data data block is a series 2 data block.

If bit 20 is set in the Request Flags field of a request message, you receive the event extra data metadata. If you want to receive both intrusion events and event extra data metadata, you must set bit 2 as well. See Request Flags on page 30. If you enable bit 23, an extended event header is included in the record.

<table>
<tr><th>Byte</th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td></tr>
<tr><td></td><td colspan="2">Header Version (1)</td><td colspan="2">Message Type (4)</td></tr>
<tr><td></td><td colspan="4">Message Length</td></tr>
<tr><td></td><td colspan="4">Record Type (111)</td></tr>
<tr><td></td><td colspan="4">Record Length</td></tr>
<tr><td></td><td colspan="4">eStreamer Server Timestamp (in events, only if bit 23 is set)</td></tr>
<tr><td></td><td colspan="4">Reserved for Future Use (in events, only if bit 23 is set)</td></tr>
<tr><td></td><td colspan="4">Event Extra Data Metadata Data Block Type (5)</td></tr>
<tr><td></td><td colspan="4">Data Block Length</td></tr>
<tr><td></td><td colspan="4">Type</td></tr>
</table>

String Block Type (0)
String Block Length
Name...
String Block Type (0)
String Block Length
Encoding

Note that the block structure includes encapsulated String block types, one of several series 2 variable length data structures introduced in Version 4.10 of the Sourcefire 3D System.

The Event Extra Data Metadata Data Block Fields table describes the fields in the Event Extra Data Metadata record.

Event Extra Data Metadata Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Event Extra Data Metadata Data Block Type	uint32	Initiates an Event Extra Data Metadata data block. This value is always 5. This block type is a series 2 block.
Event Extra Data Metadata Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Type	uint32	The type of extra data. Matches the Type field in the associated Event Extra Data record.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0. This block type is a series 2 block.
String Block Length	uint32	Number of bytes in the client application version String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the version string.
Name	string	Name of the type of event extra data, for example, XFF client (IPv6), and HTTP URI.

Event Extra Data Metadata Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a string data block for the client application URL. This value is always 0. This block type is a series 2 block.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the URL string.
Encoding	string	Encoding used for the event extra data, for example, IPv4, IPv6, or string.

Security Zone Name Record

The eStreamer service transmits metadata containing information on the name of the security zone associated with an intrusion event or connection event within a Security Zone Name record, the format of which is shown below. (Security zone information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 115, indicating a Security Zone Name record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (115)			
	Record Length			
	Security Zone Name Data Block (14)			
	Security Zone Name Data Block Length			
	Security Zone UUID			
	String Block Type (0)			
	String Block Length			
	Security Zone Name...			

The Security Zone Name Data Block Fields table describes the fields in the Security Zone Name data block.

Security Zone Name Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Security Zone Name Data Block Type	uint32	Initiates a Security Zone Name data block. This value is always 14. The block type is a series 2 block.
Security Zone Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Security Zone UUID	uint8[16]	The unique identifier for the security zone associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the security zone. This value is always 0.
String Block Length	uint32	The number of bytes included in the security zone name String data block, including eight bytes for the block type and header fields plus the number of bytes in the name.
Security Zone Name	string	The security zone name.

Interface Name Record

The eStreamer service transmits metadata containing information on the name of the interface associated with an intrusion event or connection event within an Interface Name record, the format of which is shown below. (Interface name information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 116, indicating an Interface Name record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (116)			
	Record Length			
	Interface Name Data Block (14)			
	Interface Name Data Block Length			
	Interface UUID			
	String Block Type (0)			
	String Block Length			
	Interface Name...			

The Interface Name Data Block Fields table describes the fields in the Interface Name data block.

Interface Name Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Interface Name Data Block Type	uint32	Initiates an Interface Name data block. This value is always 14. The block type is a series 2 block.
Interface Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the interface associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the interface. This value is always 0.

Interface Name Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the interface name String data block, including eight bytes for the block type and header fields plus the number of bytes in the interface name.
Interface Name	string	The interface name.

Access Control Policy Name Record

The eStreamer service transmits metadata on the name of the access control policy that triggered an intrusion event or connection event within an Access Control Policy Name record, the format of which is shown below. (Access control policy name information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 117, indicating an Access Control Policy Name record.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (117)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">Access Control Policy Name Data Block (14)</td></tr>
<tr><td></td><td colspan="32">Access Control Policy Name Data Block Length</td></tr>
<tr><td></td><td colspan="32">Access Control Policy UUID</td></tr>
<tr><td></td><td colspan="32">String Block Type (0)</td></tr>
<tr><td></td><td colspan="32">String Block Length</td></tr>
<tr><td></td><td colspan="32">Access Control Policy Name...</td></tr>
</table>

The Access Control Policy Name Data Block Fields table describes the fields in the Access Control Policy Name data block.

Access Control Policy Name Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Access Control Policy Name Data Block Type	uint32	Initiates an Access Control Policy Name data block. This value is always 14. The block type is a series 2 block.
Access Control Policy Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Access Control Policy UUID	uint8[16]	An ID number that acts as a unique identifier for the access control policy associated with the intrusion event or connection event
String Block Type	uint32	Initiates a String data block containing the name of the access control policy. This value is always 0.
String Block Length	uint32	The number of bytes included in the access control policy name String data block, including eight bytes for the block type and header fields plus the number of bytes in the access control policy name.
Access Control Policy Name	string	The access control policy name.

Access Control Rule ID Record Metadata

The eStreamer service transmits metadata containing information about the access control rule that triggered an intrusion event or connection event within an Access Control Rule ID record, the format of which is shown below. Access control rule metadata is sent when the Version 4 metadata flag—bit 20 in the

Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 119, indicating an Access Control Rule ID record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (119)			
	Record Length			
	Access Control Rule ID Data Block (15)			
	Access Control Rule ID Data Block Length			
	Access Control Rule UUID			
	Access Control Rule ID			
	String Block Type (0)			
	String Block Length			
	Access Control Rule Name...			

The Access Control Rule ID Data Block Fields table describes the fields in the Access Control Rule ID data block.

Access Control Rule ID Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Access Control Rule ID Data Block Type	uint32	Initiates an Access Control Rule ID data block. This value is always 15. The block type is a series 2 block.
Access Control Rule ID Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Access Control Rule UUID	uint8[16]	A rule ID that acts as the unique identifier for the rule in the access control policy associated with the connection event.
Access Control Rule ID	uint32	The internal identifier for the rule in the access control policy associated with the connection event.

Access Control Rule ID Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block containing the name of the access control rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the String data block, including eight bytes for the block type and header fields plus the number of bytes in the rule name.
Access Control Rule Name	string	The access control rule name.

Managed Device Record Metadata

The eStreamer service transmits metadata containing information on the managed device associated with an intrusion event within a Managed Device record, the format of which is shown below. Managed device metadata is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 123, indicating a Managed Device record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (123)			
	Record Length			
	Device ID			
	Name Length			
	Name...			

The Managed Device Record Fields table describes the fields in the Managed Device record.

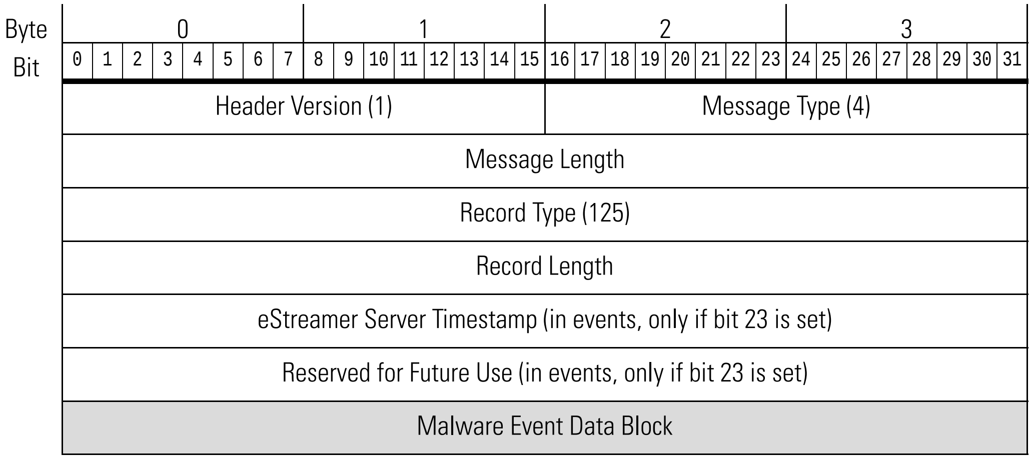
Managed Device Record Fields

Field	Data Type	Description
Device ID	uint32	ID number of the managed device.
Name Length	uint32	The number of bytes included in the name.
Name	string	The managed device name.

Malware Event Record 5.1.1+

The fields in the malware event record are shaded in the following graphic. The record type is 125.

You request malware event records by setting the malware event flag—bit 30 in the Request Flags field—in the request message with an event version of 2 and an event code of 101. See Request Flags on page 30. If you enable bit 23, an extended event header is included in the record.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (125)			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
	Malware Event Data Block			

The Malware Event Record Fields table describes each malware event record data field.

Malware Event Record Fields

FIELD	DATA TYPE	DESCRIPTION
Malware Event Data Block	variable	Indicates a malware event data block. See Malware Event Data Block 5.3+ on page 140 for more information.

Sourcefire Cloud Name Metadata

The eStreamer service transmits metadata containing information on the name of the Sourcefire cloud associated with an intrusion event or connection event within a Sourcefire Cloud Name record, the format of which is shown below. (Sourcefire Cloud name information is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 127, indicating a Sourcefire Cloud Name record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (127)			
	Record Length			
	Sourcefire Cloud Name Data Block (14)			
	Sourcefire Cloud Name Data Block Length			
	Sourcefire Cloud UUID			
	Sourcefire Cloud UUID, cont.			
	Sourcefire Cloud UUID, cont.			
	Sourcefire Cloud UUID, cont.			
	String Block Type (0)			
	String Block Length			
	Sourcefire Cloud Name...			

The Sourcefire Cloud Name Data Block Fields table describes the fields in the Sourcefire Cloud Name data block.

Sourcefire Cloud Name Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Sourcefire Cloud Name Data Block Type	uint32	Initiates a Sourcefire Cloud Name data block. This value is always 14. The block type is a series 2 block.
Sourcefire Cloud Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Sourcefire Cloud UUID	uint8[16]	A Sourcefire cloud ID number that acts as a unique identifier for the Sourcefire Cloud associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the Sourcefire Cloud. This value is always 0.
String Block Length	uint32	The number of bytes included in the Sourcefire cloud name String data block, including eight bytes for the block type and header fields plus the number of bytes in the FireAMP cloud name.
Sourcefire Cloud Name	string	The Sourcefire cloud name.

Malware Event Type Metadata

The eStreamer service transmits metadata containing malware event type information for an event within a malware event type record, the format of which is shown below. (Malware event type information is sent when the metadata flag, bit 20 in the request flags field of a request message, is set. See Request Flags on page 30.) Note that the record type field, which appears after the message length field, has a value of 128, indicating a malware event type record.

<table>
<tr><th>Byte</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (128)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">Malware Event Type ID</td></tr>
<tr><td></td><td colspan="32">Malware Event Type Length</td></tr>
<tr><td></td><td colspan="32">Malware Event Type...</td></tr>
</table>

The Malware Event Type Record Fields table describes the fields in the malware event type record.

Malware Event Type Record Fields

FIELD	DATA TYPE	DESCRIPTION
Malware Event Type ID	uint32	The malware event type ID number.
Malware Event Type Length	uint32	The number of bytes included in the malware event type.
Malware Event Type	string	The type of malware event.

Malware Event Subtype Metadata

The eStreamer service transmits metadata containing malware event subtype information for an event within a malware event subtype record, the format of which is shown below. (Malware event type information is sent when the metadata flag, bit 20 in the request flags field of a request message, is set. See

Request Flags on page 30.) Note that the record type field, which appears after the message length field, has a value of 129, indicating a malware event subtype record.

Byte	0		1		2		3	
Bit	0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15	16 17 18 19	20 21 22 23	24 25 26 27	28 29 30 31
	Header Version (1)				Message Type (4)			
	Message Length							
	Record Type (129)							
	Record Length							
	Malware Event Subtype ID							
	Malware Event Subtype Length							
	Malware Event Subtype...							

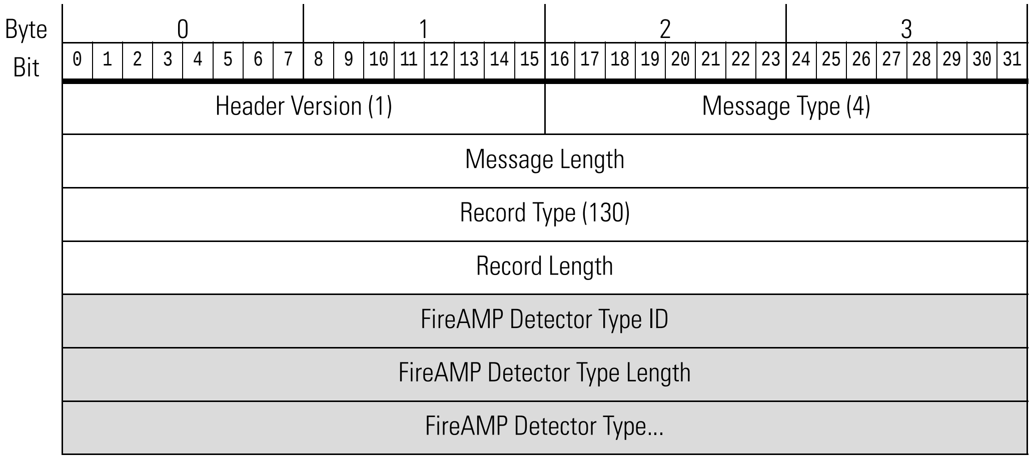
The Malware Event Subtype Record Fields table describes the fields in the malware event subtype record.

Malware Event Subtype Record Fields

Field	Data Type	Description
Malware Event Subtype ID	uint32	The malware event subtype ID number.
Malware Event Subtype Length	uint32	The number of bytes included in the malware event subtype.
Malware Event Subtype	string	The malware event subtype.

FireAMP Detector Type Metadata

The eStreamer service transmits metadata containing FireAMP detector type information for an event within a FireAMP Detector Type record, the format of which is shown below. (FireAMP detector type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 130, indicating a FireAMP detector type record.



Byte	0		1		2		3	
Bit	0 1 2 3 4 5 6 7		8 9 10 11 12 13 14 15		16 17 18 19 20 21 22 23		24 25 26 27 28 29 30 31	
	Header Version (1)				Message Type (4)			
	Message Length							
	Record Type (130)							
	Record Length							
	FireAMP Detector Type ID							
	FireAMP Detector Type Length							
	FireAMP Detector Type...							

The FireAMP Detector Type Record Fields table describes the fields in the FireAMP Detector Type record.

FireAMP Detector Type Record Fields

FIELD	DATA TYPE	DESCRIPTION
FireAMP Detector Type ID	uint32	The FireAMP detector type ID number.
FireAMP Detector Type Length	uint32	The number of bytes included in the FireAMP detector type.
FireAMP Detector Type	string	The type of FireAMP detector.

FireAMP File Type Metadata

The eStreamer service transmits metadata containing FireAMP file type information for an event within a FireAMP File Type record, the format of which is shown below. (FireAMP file type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 131, indicating a FireAMP file type record.

Byte	0		1		2		3	
Bit	0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15	16 17 18 19	20 21 22 23	24 25 26 27	28 29 30 31
	Header Version (1)				Message Type (4)			
	Message Length							
	Record Type (131)							
	Record Length							
	FireAMP File Type ID							
	FireAMP File Type Length							
	FireAMP File Type...							

The FireAMP File Type Record Fields table describes the fields in the FireAMP File Type record.

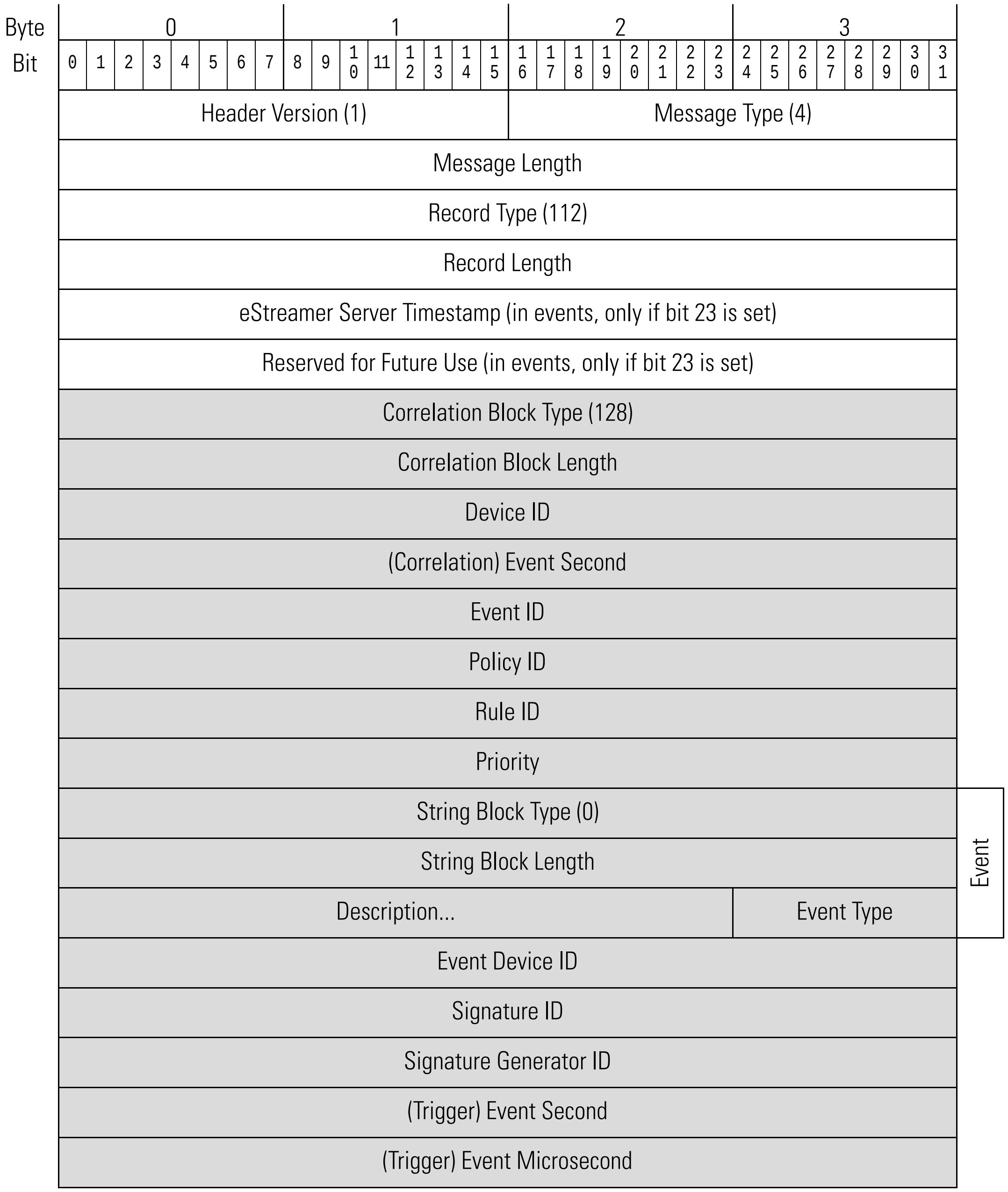
FireAMP File Type Record Fields

FIELD	DATA TYPE	DESCRIPTION
FireAMP File Type ID	uint32	The FireAMP file type ID number.
FireAMP File Type Length	uint32	The number of bytes included in the FireAMP file type.
FireAMP File Type	string	The type of detected file.

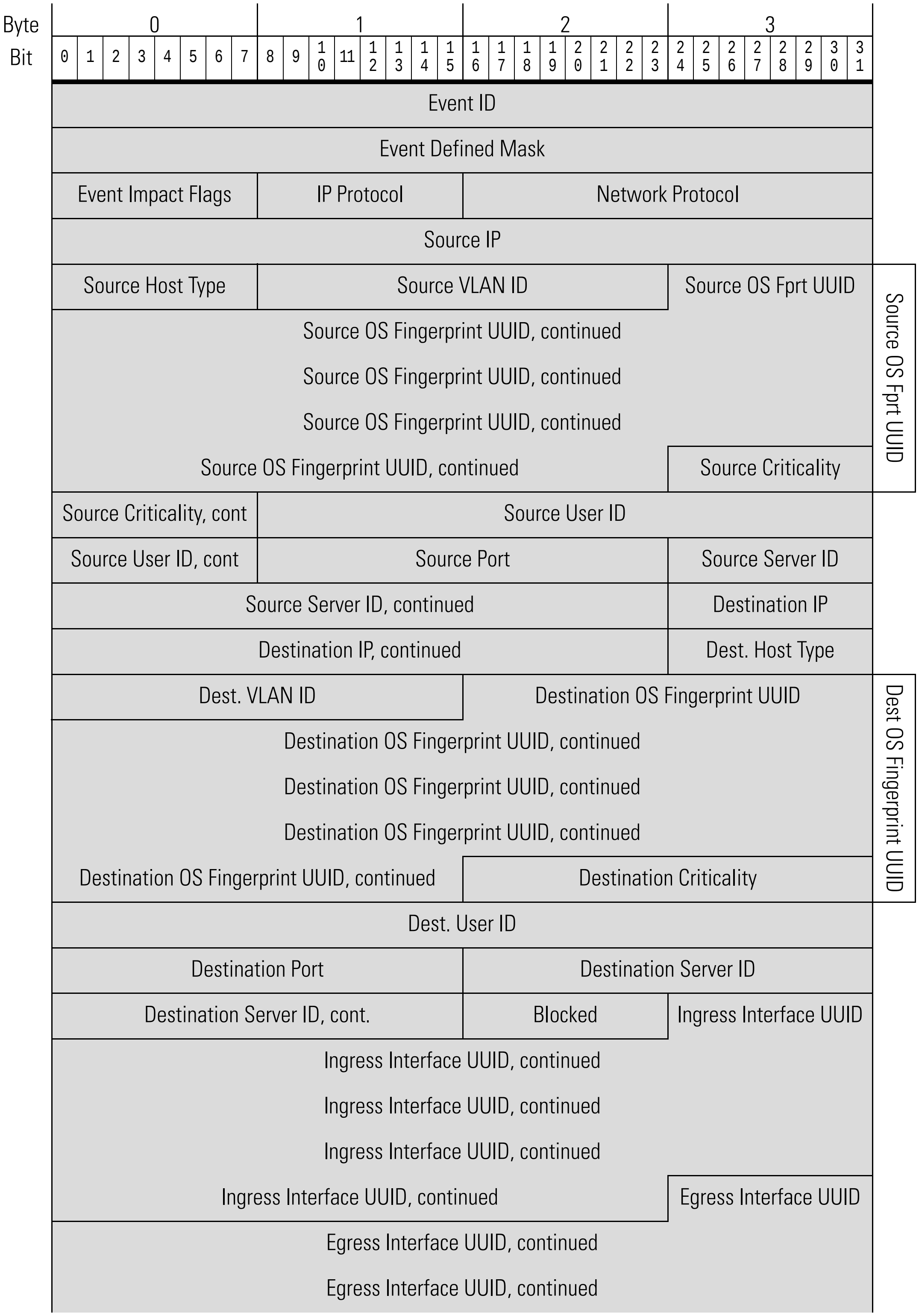
Correlation Event for 5.1+

Correlation events (called compliance events in pre-5.0 versions) contain information about correlation policy violations. This message uses the standard eStreamer message header and specifies a record type of 112, followed by a correlation data block of type 128. Data block type 128 differs from its predecessor (block type 116) in including IPv6 support.

You can request 5.1+ correlation events from eStreamer only by extended request, for which you request event type code 31 and version code 8 in the Stream Request message (see Submitting Extended Requests on page 20 for information about submitting extended requests). You can optionally enable bit 23 in the flags field of the initial event stream request message, to include the extended event header. You can also enable bit 20 in the flags field to include user metadata.



Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Header Version (1)		Message Type (4)		
	Message Length				
	Record Type (112)				
	Record Length				
	eStreamer Server Timestamp (in events, only if bit 23 is set)				
	Reserved for Future Use (in events, only if bit 23 is set)				
	Correlation Block Type (128)				
	Correlation Block Length				
	Device ID				
	(Correlation) Event Second				
	Event ID				
	Policy ID				
	Rule ID				
	Priority				
	String Block Type (0)				Event
	String Block Length				
	Description...			Event Type	
	Event Device ID				
	Signature ID				
	Signature Generator ID				
	(Trigger) Event Second				
	(Trigger) Event Microsecond				



Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Event ID				
	Event Defined Mask				
	Event Impact Flags	IP Protocol	Network Protocol		
	Source IP				
	Source Host Type	Source VLAN ID		Source OS Fprt UUID	Source OS Fprt UUID
	Source OS Fingerprint UUID, continued				
	Source OS Fingerprint UUID, continued				
	Source OS Fingerprint UUID, continued				
	Source OS Fingerprint UUID, continued			Source Criticality	
	Source Criticality, cont	Source User ID			
	Source User ID, cont	Source Port		Source Server ID	
	Source Server ID, continued			Destination IP	
	Destination IP, continued			Dest. Host Type	
	Dest. VLAN ID		Destination OS Fingerprint UUID		Dest OS Fingerprint UUID
	Destination OS Fingerprint UUID, continued				
	Destination OS Fingerprint UUID, continued				
	Destination OS Fingerprint UUID, continued				
	Destination OS Fingerprint UUID, continued		Destination Criticality		
	Dest. User ID				
	Destination Port		Destination Server ID		
	Destination Server ID, cont.		Blocked	Ingress Interface UUID	
	Ingress Interface UUID, continued				
	Ingress Interface UUID, continued				
	Ingress Interface UUID, continued				
	Ingress Interface UUID, continued			Egress Interface UUID	
	Egress Interface UUID, continued				
	Egress Interface UUID, continued				

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Egress Interface UUID, continued			
	Egress Interface UUID, continued			Ingress Zone UUID
	Ingress Zone UUID			
	Ingress Zone UUID, continued			
	Ingress Zone UUID, continued			
	Ingress Zone UUID, continued			Egress Zone UUID
	Egress Zone UUID			
	Egress Zone UUID, continued			
	Egress Zone UUID, continued			
	Egress Zone UUID, continued			Source IPv6 Address
	Source IPv6 Address			
	Source IPv6 Address, continued			
	Source IPv6 Address continued			
	Source IPv6 Address, continued			Destination IPv6 Address
	Destination IPv6 Address			
	Destination IPv6 Address, continued			
	Destination IPv6 Address, continued			
	Destination IPv6 Address, continued			

Note that the record structure includes a String block type, which is a block in series 1. For information about series 1 blocks, see Understanding Discovery (Series 1) Blocks on page 224.

Correlation Event 5.1+ Data Fields

FIELD	DATA TYPE	DESCRIPTION
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of 128. See Understanding Discovery (Series 1) Blocks on page 224.
Correlation Block Length	uint32	Length of the correlation data block, which includes 8 bytes for the correlation block type and length plus the correlation data that follows.
Device ID	uint32	Internal identification number of the managed device or Defense Center that generated the correlation event. A value of zero indicates the Defense Center. You can obtain managed device names by requesting Version 3 metadata. See Managed Device Record Metadata on page 99 for more information.
(Correlation) Event Second	uint32	UNIX timestamp indicating the time that the correlation event was generated (in seconds from 01/01/1970).
Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Priority	uint32	Priority assigned to the event. This is an integer value from 0 to 5.
String Block Type	uint32	Initiates a string data block that contains the correlation violation event description. This value is always set to 0. For more information about string blocks, see String Data Block on page 237.

Correlation Event 5.1+ Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.
Description	string	Description of the correlation event.
Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, host discovery, or user event: • 1 — intrusion • 2 — host discovery • 3 — user
Event Device ID	uint32	Identification number of the device that generated the event that triggered the correlation event. You can obtain device name by requesting Version 3 metadata. See Managed Device Record Metadata on page 99 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is 0.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Sourcefire 3D System preprocessor or rules engine that generated the event.
(Trigger) Event Second	uint32	UNIX timestamp indicating the time of the event that triggered the correlation policy rule (in seconds from 01/01/1970).
(Trigger) Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the Sourcefire device.
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See the Event Defined Values table on page 115 for a list of each bit value.

Correlation Event 5.1+ Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Event Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
IP Protocol	uint8	Identifier of the IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP	uint8[4]	IP address of the source host in the event, in IP address octets.

Correlation Event 5.1+ Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Source Host Type	uint8	Source host's type: • 0 — Host • 1 — Router • 2 — Bridge
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	IP address of the destination host associated with the policy violation (if applicable). This value will be 0 if there is no destination IP address.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.

Correlation Event 5.1+ Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Service ID	uint32	Identification number for the server running on the source host.
Blocked	uint8	Value indicating what happened to the packet that triggered the intrusion event. • 0 — Intrusion event not dropped • 1 — Intrusion event was dropped (drop when deployment is inline, switched, or routed) • 2 — The packet that triggered the event would have been dropped, if the intrusion policy had been applied to a device in inline, switched, or routed deployment.
Ingress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the ingress interface associated with correlation event.
Egress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the egress interface associated with correlation event.
Ingress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the ingress security zone associated with correlation event.

Correlation Event 5.1+ Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Egress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the egress security zone associated with correlation event.
Source IPv6 Address	uint8[16]	IP address of the source host in the event, in IPv6 address octets.
Destination IPv6 Address	uint8[16]	IP address of the destination host in the event, in IPv6 address octets.

The Event Defined Values table describes each Event Defined Mask value.

Event Defined Values

DESCRIPTION	MASK VALUE
Event Impact Flags	`0x00000001`
IP Protocol	`0x00000002`
Network Protocol	`0x00000004`
Source IP	`0x00000008`
Source Host Type	`0x00000010`
Source VLAN ID	`0x00000020`
Source Fingerprint ID	`0x00000040`
Source Criticality	`0x00000080`
Source Port	`0x00000100`
Source Server	`0x00000200`
Destination IP	`0x00000400`
Destination Host Type	`0x00000800`
Destination VLAN ID	`0x00001000`
Destination Fingerprint ID	`0x00002000`

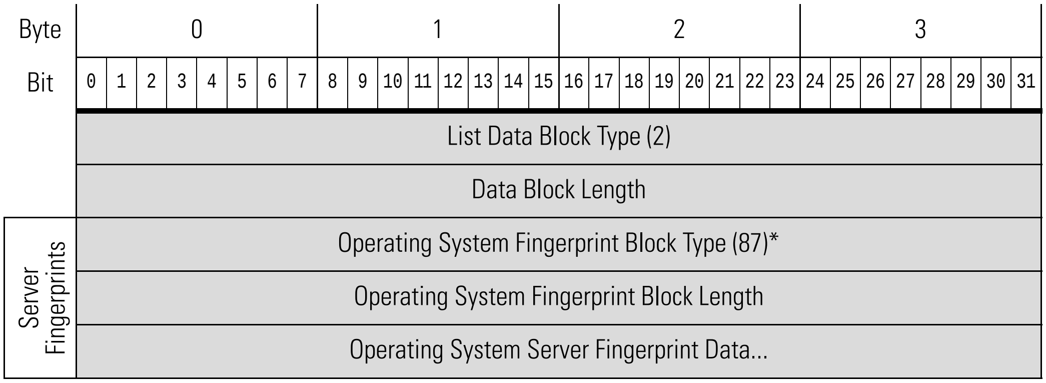
Event Defined Values (Continued)

DESCRIPTION	MASK VALUE
Destination Criticality	`0x00004000`
Destination Port	`0x00008000`
Destination Server	`0x00010000`
Source User	`0x00020000`
Destination User	`0x00040000`

Understanding Series 2 Data Blocks

Beginning in version 4.10.0, the eStreamer service uses a second series of data blocks to package certain records such as intrusion event extra data. See the Series 2 Block Types table on page 117 for a list of all block types in the series. Series 2 blocks, like series 1 blocks, support variable-length fields and hierarchies of nested blocks. The series 2 block types include primitive blocks that provide the same mechanism for encapsulating nested inner blocks as the series 1 primitive block types. However, series 2 blocks and series 1 blocks have separate numbering systems.

The following example shows the how primitive blocks are used. The list data block (series 2 block type 31) defines an array of operating system fingerprints (each of which is a type 87 block itself with variable length). The overall type 31 data block length is self-describing via the Data Block Length field, which contains the length of the data portion of the message, excluding the 8 bytes in the block type and block length fields.



In the Series 2 Block Types table below, the Data Block Status field indicates whether the block is current (the latest version) or legacy (used in an older version and can still be requested through eStreamer).

Series 2 Block Types

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
0	String	Current	Encapsulates variable string data. See String Data Block on page 121 for more information.
1	BLOB	Current	Encapsulates binary data and is used specifically for banners. See BLOB Data Block on page 122 for more information.
2	List	Current	Encapsulates a list of other data blocks. See List Data Block on page 123 for more information.
3	Generic List	Current	Encapsulates a list of other data blocks. For deserialization, it is the equivalent of the List data block. See Generic List Data Block on page 124 for more information.
4	Event Extra Data	Current	Contains intrusion event extra data. See Intrusion Event Extra Data Record on page 89 for more information.
5	Extra Data Type	Current	Contains extra data metadata. See Intrusion Event Extra Data Metadata on page 91 for more information.
14	UUID String Mapping	Current	Block used by various metadata messages to map UUID values to descriptive strings. See UUID String Mapping Data Block on page 125.
15	Access Control Policy Rule ID Metadata	Current	Contains metadata for access control rules. See Access Control Policy Rule ID Metadata Block on page 126.

Series 2 Block Types (Continued)

Type	Content	Data Block Status	Description
16	Malware Event	Legacy	Contains information on malware events, such as the malware detected or quarantined within a FireAMP cloud, the detection method, and hosts and users affected by the malware. See Malware Event Data Block 5.1 on page 492. Deprecated by block 24, Malware Event Data Block 5.3+ on page 140.
19	ICMP Type Data Block	Current	Contains metadata describing ICMP types. See ICMP Type Data Block on page 128.
20	ICMP Code Data Block	Current	Contains metadata describing ICMP codes. See ICMP Code Data Block on page 129.
21	Access Control Policy Rule Reason Data Block	Current	Contains information explaining access control policy rule reasons. See Access Control Policy Rule Reason Data Block on page 131.
22	IP Reputation Category Data Block	Current	Contains information on IP reputation categories explaining why an IP address was blocked. See IP Reputation Category Data Block on page 132.
23	File Event	Legacy	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.1.1.x on page 619. It is superseded by block 32, Access Control Policy Rule ID Metadata Block on page 126.

Series 2 Block Types (Continued)

Type	Content	Data Block Status	Description
24	Malware Event	Legacy	Contains information on malware events, such as the malware detected or quarantined within a FireAMP cloud, the detection method, and hosts and users affected by the malware. See Malware Event Data Block 5.1.1.x on page 497. Deprecates block 16, Malware Event Data Block 5.1 on page 492. Deprecated by block 33, Malware Event Data Block 5.3+ on page 140.
25	Intrusion Event	Legacy	Contains information on intrusion events, including information to match intrusion events with connection and malware events. See Intrusion Event Record 5.3+ on page 70. Deprecated by Intrusion Event Record 5.3+ on page 70.
26	File Event SHA Hash	Legacy	Contains the SHA hash and name of files that have been identified as containing malware. See File Event SHA Hash for 5.1.1-5.2.x on page 628. Deprecated by block 40.
27	Rule Documentation Data Block	Current	Contains information about rules used to generate events. See Rule Documentation Data Block for 5.2+ on page 151 for more information.
28	Geolocation Data Block	Current	Contains country codes and associated country name. See IOC Name Data Block for 5.3+ on page 160.
32	File Event	Legacy	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.2.x on page 623. It deprecates File Event for 5.1.1.x on page 619. Deprecated by block 38.

Series 2 Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
33	Malware Event	Current	Contains information on malware events, such as the malware detected or quarantined within a FireAMP cloud, the detection method, and hosts and users affected by the malware. See Malware Event Data Block 5.2.x on page 505. Deprecates block 24, Malware Event Data Block 5.1.1.x on page 497. Deprecated by block 35
34	Intrusion Event	Legacy	Contains information on intrusion events, including information to match intrusion events with connection and malware events. See Intrusion Event Record 5.2.x on page 478. Deprecated block 25. Deprecated by block 41.
35	Malware Event	Current	Contains information on malware events, including IOC information. See Malware Event Data Block 5.3+ on page 140. Deprecates block 33, Malware Event Data Block 5.2.x on page 505.
38	File Event	Current	Contains information on file events, such as the source, SHA hash, and the disposition of the file. See File Event for 5.3+ on page 133. It deprecates block 32.
39	IOC Name Data Block	Current	Contains information about IOCs. See IOC Name Data Block for 5.3+ on page 160
40	File Event SHA Hash	Current	Contains the SHA hash and name of files that have been identified as containing malware. See File Event SHA Hash for 5.3+ on page 149. Deprecated block 26
41	Intrusion Event	Current	Contains information on intrusion events, including information to match intrusion events with IOCs. See Intrusion Event Record 5.3+ on page 70. Deprecated block 34.

Series 2 Primitive Data Blocks

Both series 2 and series 1 blocks include a set of primitives that are used to encapsulate lists of variable-length blocks as well as variable-length strings and BLOBs within messages. These primitive blocks have the standard eStreamer block header discussed above in Data Block Header on page 46, but they appear only within other data blocks. Any number can be included in a given block type. For details on the structure of these blocks, see the following:

- String Data Block on page 121
- BLOB Data Block on page 122
- List Data Block on page 123
- Generic List Data Block on page 124

String Data Block

The eStreamer service uses the String data block to send string data in messages. These blocks commonly appear within other data blocks to identify, for example, operating system or server names.

Empty String data blocks (containing no data, only the header fields) have a block length of 8. eStreamer uses an empty String data block when it has no content for a string value, as might happen, for example, in the OS vendor string field in an Operating System data block when the vendor of the operating system is unknown.

The String data block has a block type of 0 in the series 2 group of blocks.

IMPORTANT! Strings returned in this data block are not always null-terminated (that is, the string characters are not always followed by a 0).

The following diagram shows the format of the String data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Data Block Type (0)</td></tr>
<tr><td></td><td colspan="32">Data Block Length</td></tr>
<tr><td></td><td colspan="32">String Data...</td></tr>
</table>

The String Block Fields table describes the fields of the String data block.

String Block Fields

FIELD	DATA TYPE	DESCRIPTION
Data Block Type	uint32	Initiates a String data block. This value is always 0.
Data Block Length	uint32	Combined length in bytes of the string data block header and string data.
String Data	string	Contains the string data and may contain a terminating character (null byte) at the end of the string.

BLOB Data Block

The eStreamer service uses the BLOB data block to convey binary data. For example, host discovery records use the BLOB block to hold captured server banners. The BLOB data block has a block type of 1 in the series 2 group of blocks.

The following diagram shows the format of the BLOB data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Data Block Type (1)</td></tr>
<tr><td></td><td colspan="32">Data Block Length</td></tr>
<tr><td></td><td colspan="32">Binary Data...</td></tr>
</table>

The BLOB Data Block Fields table describes the fields of the BLOB data block.

BLOB Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Data Block Type	uint32	Initiates a BLOB data block. This value is always 1.
Data Block Length	uint32	Number of bytes in the BLOB data block, including eight bytes for the BLOB block type and length fields, plus the length of the binary data that follows.
Binary Data	variable	Contains binary data such as a server banner.

List Data Block

The eStreamer service uses the List data block to encapsulate a list of data blocks. For example, eStreamer can use the List data block to send a list of TCP servers, each of which is itself a data block. The List data block has a block type of 2 in the series 2 group of blocks.

The following diagram shows the basic format of a List data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Block Type (2)</td></tr>
<tr><td></td><td colspan="32">Block Length</td></tr>
<tr><td></td><td colspan="32">Encapsulated Data Blocks...</td></tr>
</table>

The List Data Fields table describes the fields of the List data block.

List Data Fields

FIELD	DATA TYPE	DESCRIPTION
Block Type	uint32	Initiates a List data block. This value is always 2.
Block Length	uint32	Number of bytes in the List block and encapsulated data. For example, if there were three Sub-Server data blocks included in the list, the value here would include the total number of bytes in the Sub-Server blocks, plus eight bytes for the List block header.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the list block length.

Generic List Data Block

The eStreamer service uses the Generic List data block to encapsulate a list of data blocks. For example, the Host Profile data block contains information about multiple client applications and uses the Generic List block to embed a list of Client Application data blocks in the message. The Generic List data block has a block type of 3 in the series 2 group of blocks.

The following diagram shows the basic structure of a Generic List data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Data Block Type (3)</td></tr>
<tr><td></td><td colspan="32">Data Block Length</td></tr>
<tr><td></td><td colspan="32">Encapsulated Data Blocks...</td></tr>
</table>

The Generic List Data Block table describes the fields of the Generic List data block.

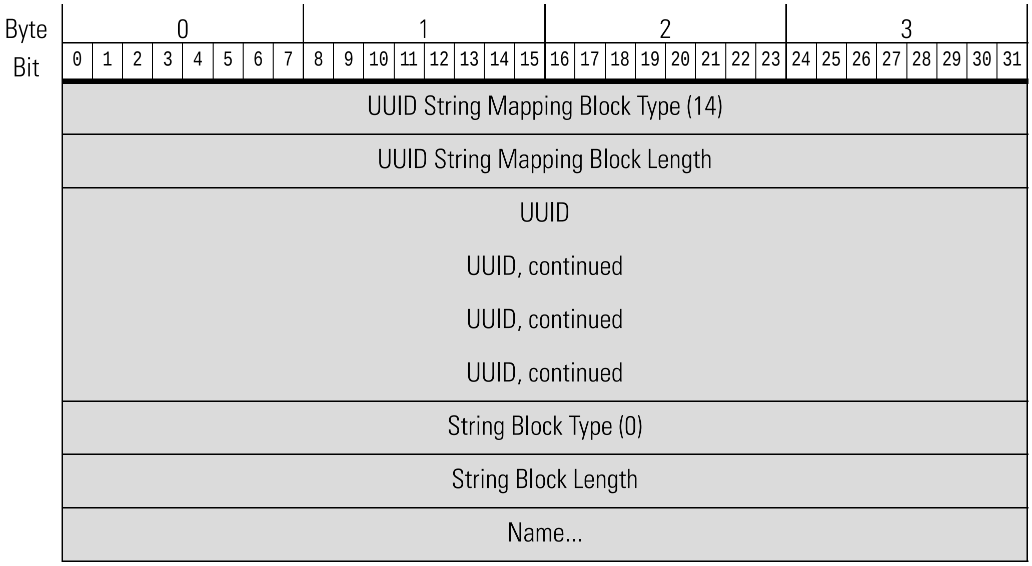
Generic List Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
Data Block Type	uint32	Initiates a Generic List data block. This value is always 3.
Data Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the total number of bytes in all of the encapsulated data blocks.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the Generic List block length.

UUID String Mapping Data Block

The eStreamer service uses the UUID String Mapping data block in various metadata messages to map UUID values to descriptive strings. The UUID String Mapping data block has a block type of 14 in series 2.

The following diagram shows the structure of the UUID String Mapping data block.



UUID String Mapping Block Type (14)

UUID String Mapping Block Length

UUID

UUID, continued

UUID, continued

UUID, continued

String Block Type (0)

String Block Length

Name...

The UUID String Mapping Data Block Fields table describes the fields in the UUID String Mapping data block.

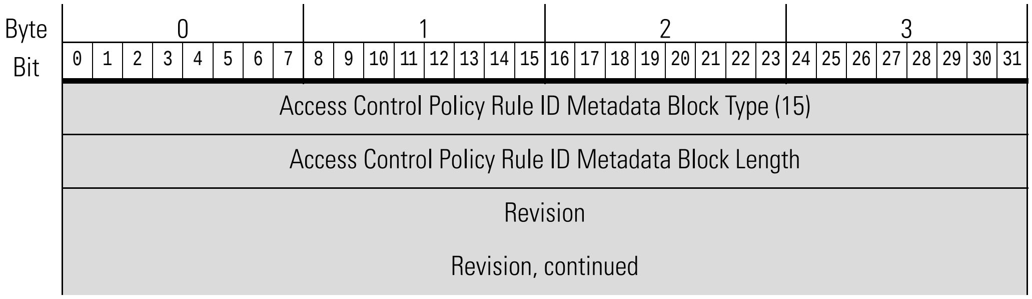
UUID String Mapping Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
UUID String Mapping Block Type	uint32	Initiates a UUID String Mapping block. This value is always 14.
UUID String Mapping Block Length	uint32	Total number of bytes in the UUID String Mapping block, including eight bytes for the UUID String Mapping block type and length fields, plus the number of bytes of data that follows.
UUID	uint8[16]	The unique identifier for the event or other object the UUID identifies.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the UUID. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The descriptive name.

Access Control Policy Rule ID Metadata Block

The eStreamer service uses the Access Control Policy Rule ID metadata block to contain information about access control policy rule IDs. This data block has a block type of 15 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Access Control Policy Rule ID Metadata Block Type (15)

Access Control Policy Rule ID Metadata Block Length

Revision

Revision, continued

	Revision, continued Revision, continued
	Rule ID
Name	String Block Type (0)
	String Block Length
	Name...

The Access Control Policy Rule ID Metadata Block Fields table describes the fields in the Access Control Policy Rule ID Metadata block.

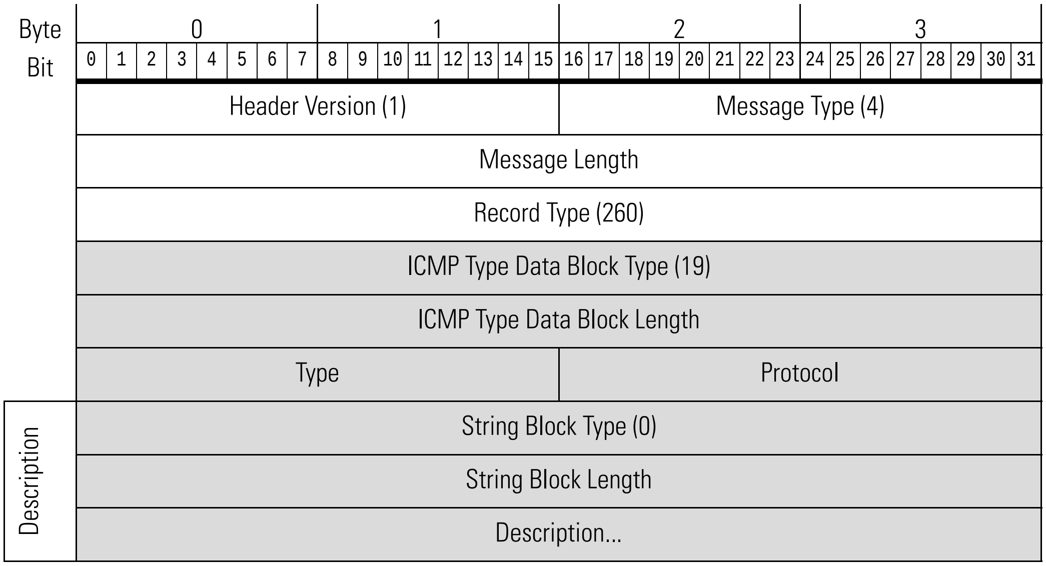
Access Control Policy Rule ID Metadata Block Fields

Field	Data Type	Description
Access Control Policy Rule ID Metadata Block Type	uint32	Initiates a Access Control Policy Rule ID Metadata block. This value is always 15.
Access Control Policy Rule ID Metadata Block Length	uint32	Total number of bytes in the Access Control Policy Rule ID block, including eight bytes for the Access Control Policy Rule ID metadata block type and length fields, plus the number of bytes of data that follows.
Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event.
Rule ID	uint32	Internal identifier for the rule that triggered the event.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control policy rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The descriptive name of the access control policy rule.

ICMP Type Data Block

The eStreamer service uses the ICMP Type data block to contain information about ICMP Types. This data block has a record type of 260, and a block type of 19 in series 2.

The following diagram shows the structure of the ICMP Type data block.



The ICMP Type Data Block Fields table describes the fields in the ICMP Type data block.

ICMP Type Data **Block Fields**

Field	Data Type	Description
ICMP Type Data Block Type	uint32	Initiates an ICMP Type data block. This value is always 19.
ICMP Type Data Block Length	uint32	Total number of bytes in the ICMP Type data block, including eight bytes for the ICMP Type data block type and length fields, plus the number of bytes of data that follows.
Type	uint16	The ICMP type of the event.

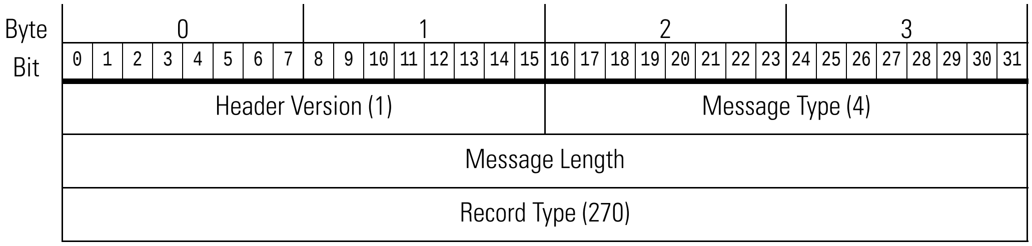
ICMP Type Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Protocol	uint16	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.
String Block Type	uint32	Initiates a String data block containing the description of the ICMP type. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the ICMP type for the event.

ICMP Code Data Block

The eStreamer service uses the ICMP Code data block to contain information about access control policy rule IDs. This data block has a record type of 270, and block type of 20 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.



	ICMP Code Data Block Type (20)	
	ICMP Code Data Block Length	
	Code	Type
Description	Protocol	String Block Type (0)
	String Block Type (0), continued	String Block Length
	String Block Length, continued	Description...

The ICMP Code Data Block Fields table describes the fields in the ICMP Code data block.

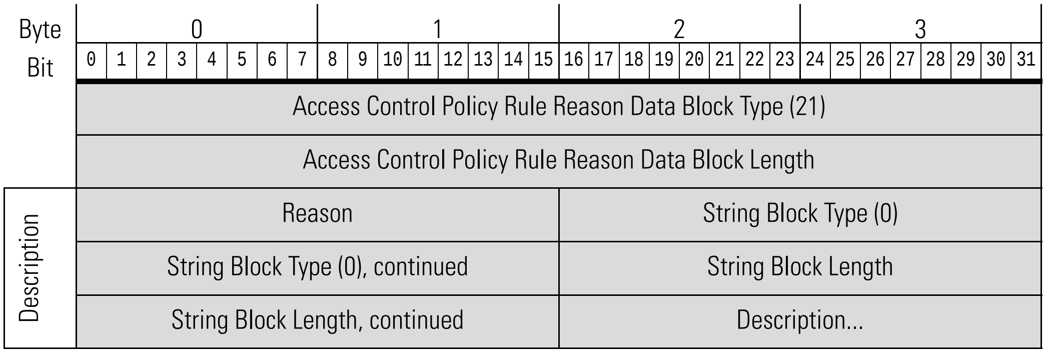
ICMP Code Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
ICMP Code Data Block Type	uint32	Initiates a ICMP Code data block. This value is always 20.
ICMP Code Data Block Length	uint32	Total number of bytes in the ICMP Code data block, including eight bytes for the ICMP Code data block type and length fields, plus the number of bytes of data that follows.
Code	uint16	The ICMP code of the event.
Type	uint16	The ICMP type of the event.
Protocol	uint16	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.
String Block Type	uint32	Initiates a String data block containing the description of the ICMP code. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the ICMP code for the event.

Access Control Policy Rule Reason Data Block

The eStreamer service uses the Access Control Rule Policy Rule Reason Data block to contain information about access control policy rule IDs. This data block has a block type of 21 in series 2.

The following diagram shows the structure of the Access Control Policy Rule ID metadata block.



	Byte 0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Access Control Policy Rule Reason Data Block Type (21)			
	Access Control Policy Rule Reason Data Block Length			
Description	Reason		String Block Type (0)	
	String Block Type (0), continued		String Block Length	
	String Block Length, continued		Description...	

The Access Control Policy Rule Reason Data Block Fields table describes the fields in the Access Control Policy Rule ID metadata block.

Access Control Policy Rule Reason Data **Block Fields**

FIELD	DATA TYPE	DESCRIPTION
Access Control Policy Rule Reason Data Block Type	uint32	Initiates an Access Control Policy Rule Reason data block. This value is always 21.
Access Control Policy Rule Reason Data Block Length	uint32	Total number of bytes in the Access Control Policy Rule Reason data block, including eight bytes for the Access Control Policy Rule Reason data block type and length fields, plus the number of bytes of data that follows.
Reason	uint16	The number of the reason for the rule that triggered the event.
String Block Type	uint32	Initiates a String data block containing the description of the access control policy rule reason. This value is always 0.

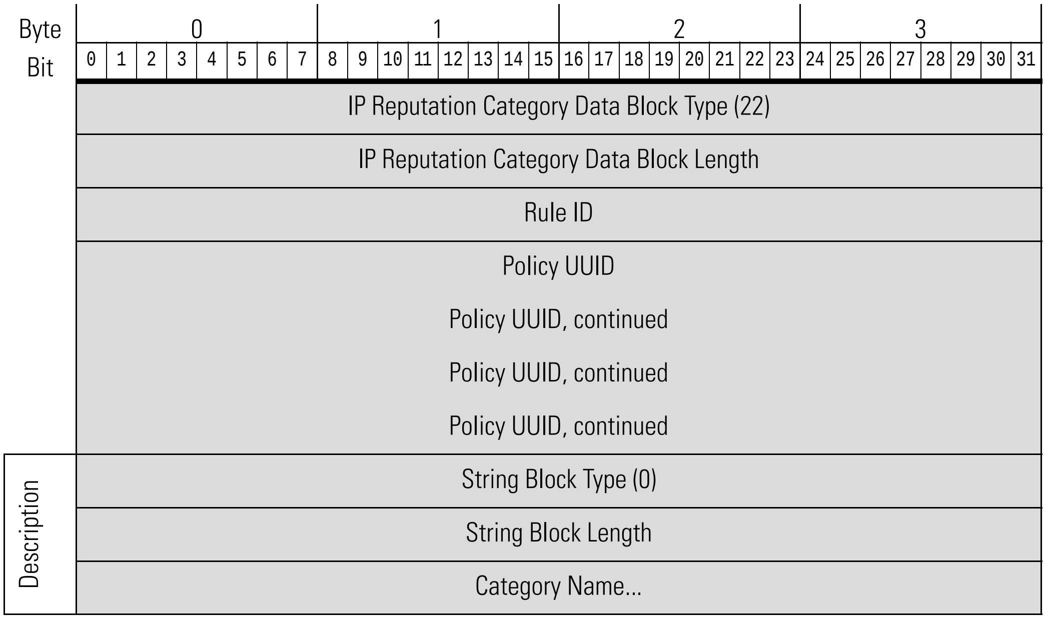
Access Control Policy Rule Reason Data **Block Fields (Continued)**

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the reason for the rule.

IP Reputation Category Data Block

The eStreamer service uses the IP Reputation Category Data block to contain information about rule reputation categories. This data block has a block type of 22 in series 2.

The following diagram shows the structure of the IP Reputation Category data block.



The IP Reputation Category Data Block Fields table describes the fields in the IP Reputation Category Data Block.

IP Reputation Category Data Block Fields

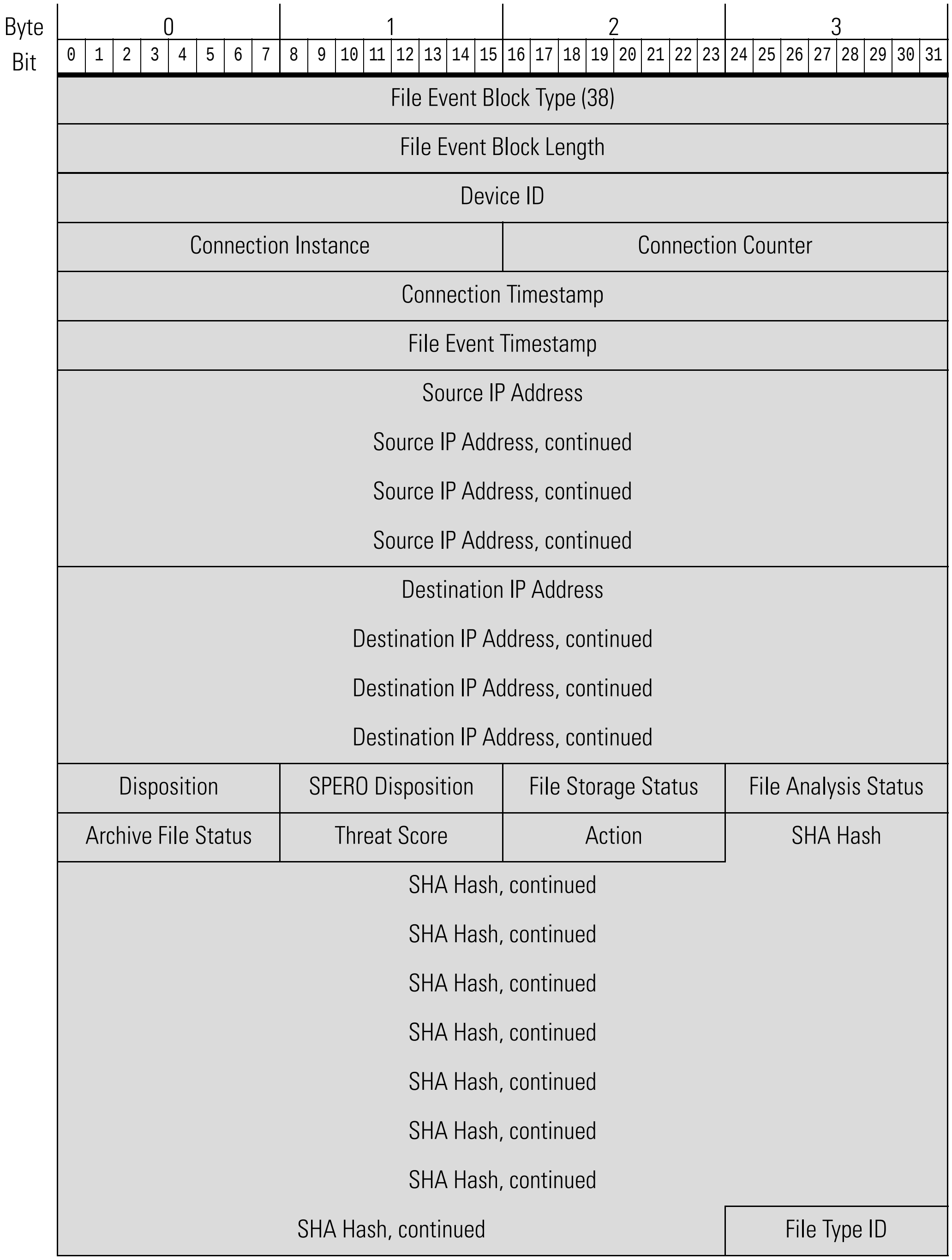
FIELD	DATA TYPE	DESCRIPTION
IP Reputation Category Data Block Type	uint32	Initiates a IP Reputation Category data block. This value is always 22.
IP Reputation Category Data Block Length	uint32	Total number of bytes in the IP Reputation Category data block, including eight bytes for the IP Reputation Category data block type and length fields, plus the number of bytes of data that follows.
Rule ID	uint32	Internal identifier for the rule that triggered the event.
Policy UUID	uint8[16]	UUID of the policy that triggered the event.
String Block Type	uint32	Initiates a String data block containing the description of the IP Reputation Category. This value is always 0.
String Block Length	uint32	The number of bytes included in the Category Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Category Name field.
Category Name	string	Name of the category for the rule.

File Event for 5.3+

The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 38 in the series 2 group of blocks. It supersedes block type 32. New fields have been added to track dynamic file analysis and file storage.

You request file event records by setting the file event flag—bit 30 in the Request Flags field—in the request message with an event version of 3 and an event code of 111. See Request Flags on page 30. If you enable bit 23, an extended event header is included in the record.

The following graphic shows the structure of the File Event data block.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	File Event Block Type (38)			
	File Event Block Length			
	Device ID			
	Connection Instance		Connection Counter	
	Connection Timestamp			
	File Event Timestamp			
	Source IP Address			
	Source IP Address, continued			
	Source IP Address, continued			
	Source IP Address, continued			
	Destination IP Address			
	Destination IP Address, continued			
	Destination IP Address, continued			
	Destination IP Address, continued			
	Disposition	SPERO Disposition	File Storage Status	File Analysis Status
	Archive File Status	Threat Score	Action	SHA Hash
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			File Type ID

File Name	File Type ID, cont.			String Block Type (0)
	String Block Type (0), cont.			String Block Length
	String Block Length, cont.			File Name...
	File Size			
	File Size, continued			
	Direction	Application ID		
	App ID, cont.	User ID		
URI	User ID, cont.	String Block Type (0)		
	String Block Type (0), cont.	String Block Length		
	String Block Length, cont.	URI...		
Signature	String Block Type (0)			
	String Block Length			
	Signature...			
	Source Port		Destination Port	
	Protocol	Access Control Policy UUID		
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	AC Pol UUID, cont.	Source Country		Dst. Country
	Dst. Country, cont.	Web Application ID		
	Web App. ID, cont.	Client Application ID		
	Client App. ID, cont.			

The File Event Data Block Fields table describes the fields in the file event data block.

File Event Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
File Event Block Type	uint32	Initiates whether file event data block. This value is always 23.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.

File Event Data Block Fields (Continued)

Field	Data Type	Description
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — UNKNOWN — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — UNAVAILABLE — The software was unable to send a request to the Sourcefire cloud for a disposition, or the Sourcefire cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE — The file matches a user-defined hash, and is treated in a fashion designated by the user.
SPERO Disposition	uint8	Indicates whether the SPERO signature was used in file analysis. If the value is 1, 2, or 3, SPERO analysis was used. If there is any other value SPERO analysis was not used.
File Storage Status	uint8	The storage status of the file. Possible values are: • 1 — File Stored • 2 — File Stored • 3 — Unable to Store File • 4 — Unable to Store File • 5 — Unable to Store File • 6 — Unable to Store File • 7 — Unable to Store File • 8 — File Size is Too Large • 9 — File Size is Too Small • 10 — Unable to Store File • 11 — File Not Stored, Disposition Unavailable

File Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
File Analysis Status	uint8	Indicates whether the file was sent for dynamic analysis. Possible values are: • `1` — Sent for Analysis • `2` — Sent for Analysis • `4` — Sent for Analysis • `5` — Failed to Send • `6` — Failed to Send • `7` — Failed to Send • `8` — Failed to Send • `9` — File Size is Too Small • `10` — File Size is Too Large • `11` — Sent for Analysis • `12` — Analysis Complete • `13` — Failure (Network Issue) • `14` — Failure (Rate Limit) • `15` — Failure (File Too Large) • `16` — Failure (File Read Error) • `17` — Failure (Internal Library Error) • `19` — File Not Sent, Disposition Unavailable • `20` — Failure (Cannot Run File) • `21` — Failure (Analysis Timeout) • `22` — Sent for Analysis • `23` — File Not Supported
Archive File Status	uint8	This is always `0`.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.

File Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type. The meaning of this field is transmitted in the metadata with this event. See FireAMP File Type Metadata on page 105 for more information.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.

File Event Data Block Fields (Continued)

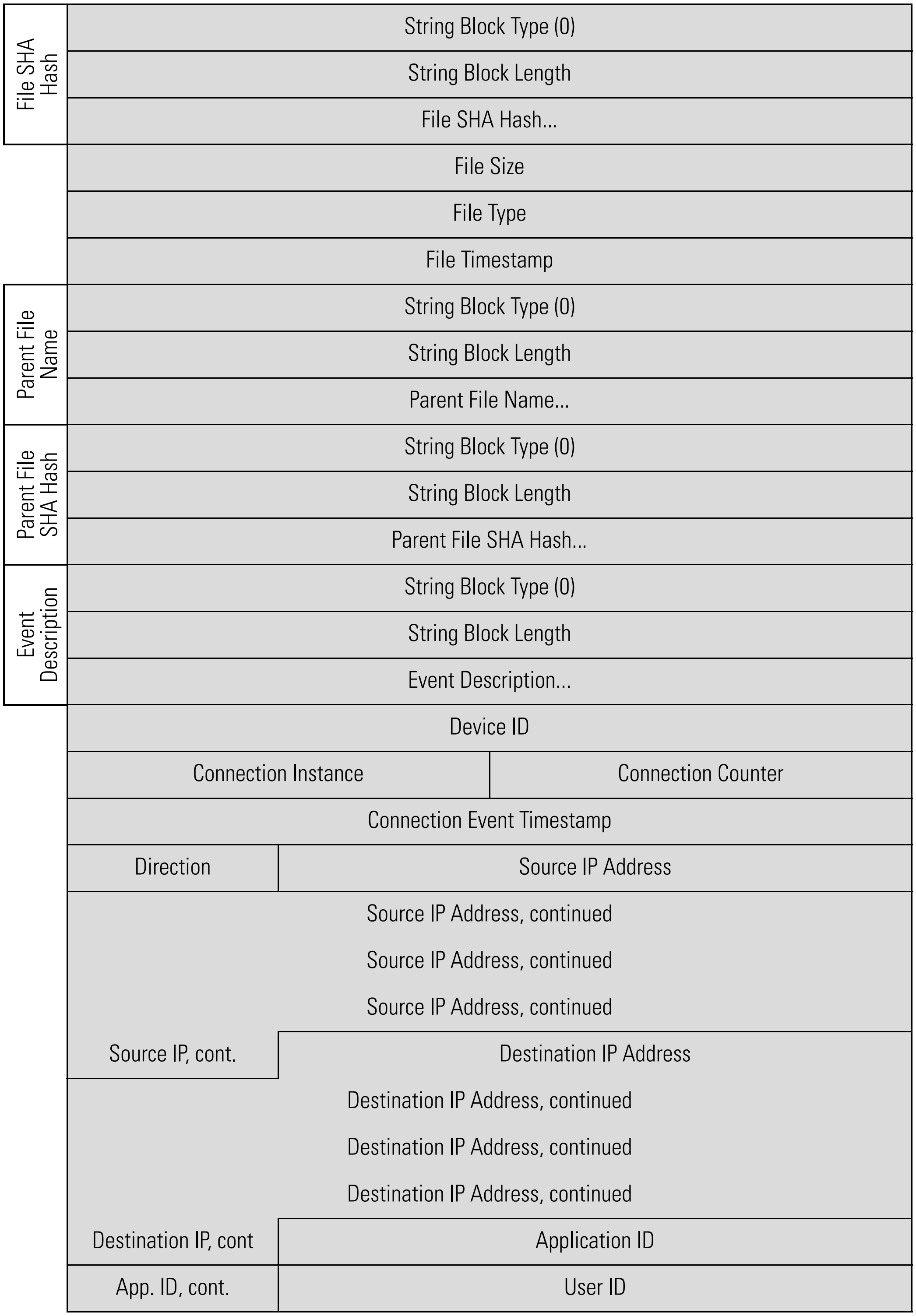
FIELD	DATA TYPE	DESCRIPTION
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • `1` — ICMP • `4` — IP • `6` — TCP • `17` — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.

Malware Event Data Block 5.3+

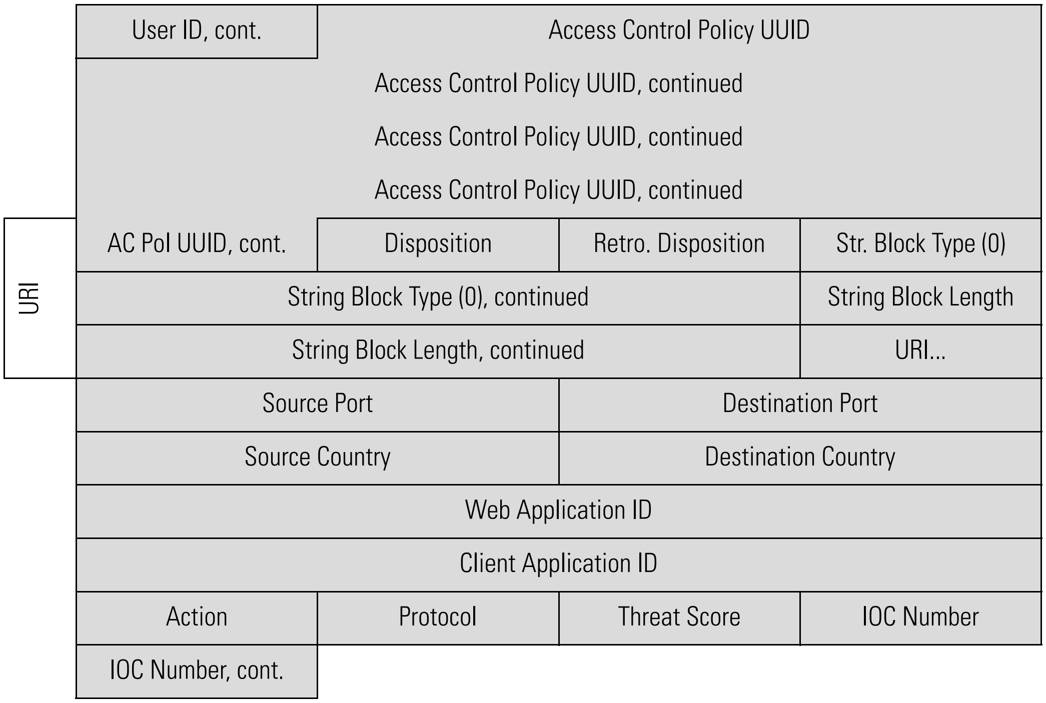
The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 35 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 4 and an event code of 101.

The following graphic shows the structure of the malware event data block:

<table>
<tr><th>Byte</th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td></tr>
<tr><td></td><td colspan="4">Malware Event Block Type (35)</td></tr>
<tr><td></td><td colspan="4">Malware Event Block Length</td></tr>
<tr><td></td><td colspan="4">Agent UUID
Agent UUID, continued
Agent UUID, continued
Agent UUID, continued</td></tr>
<tr><td></td><td colspan="4">Cloud UUID
Cloud UUID, continued
Cloud UUID, continued
Cloud UUID, continued</td></tr>
<tr><td></td><td colspan="4">Malware Event Timestamp</td></tr>
<tr><td></td><td colspan="4">Event Type ID</td></tr>
<tr><td></td><td colspan="4">Event Subtype ID</td></tr>
<tr><td rowspan="3">Detection Name</td><td>Detector ID</td><td colspan="3">String Block Type (0)</td></tr>
<tr><td>String Block Type (0), cont.</td><td colspan="3">String Block Length</td></tr>
<tr><td>String Block Length, cont.</td><td colspan="3">Detection Name...</td></tr>
<tr><td rowspan="3">User</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">User...</td></tr>
<tr><td rowspan="3">File Name</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">File Name...</td></tr>
<tr><td rowspan="3">File Path</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">File Path...</td></tr>
</table>



File SHA Hash	String Block Type (0)	
	String Block Length	
	File SHA Hash...	
	File Size	
	File Type	
	File Timestamp	
Parent File Name	String Block Type (0)	
	String Block Length	
	Parent File Name...	
Parent File SHA Hash	String Block Type (0)	
	String Block Length	
	Parent File SHA Hash...	
Event Description	String Block Type (0)	
	String Block Length	
	Event Description...	
	Device ID	
	Connection Instance	Connection Counter
	Connection Event Timestamp	
	Direction	Source IP Address
	Source IP Address, continued	
	Source IP Address, continued	
	Source IP Address, continued	
	Source IP, cont.	Destination IP Address
	Destination IP Address, continued	
	Destination IP Address, continued	
	Destination IP Address, continued	
	Destination IP, cont	Application ID
	App. ID, cont.	User ID



	User ID, cont.	Access Control Policy UUID		
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
URI	AC Pol UUID, cont.	Disposition	Retro. Disposition	Str. Block Type (0)
	String Block Type (0), continued			String Block Length
	String Block Length, continued			URI...
	Source Port		Destination Port	
	Source Country		Destination Country	
	Web Application ID			
	Client Application ID			
	Action	Protocol	Threat Score	IOC Number
	IOC Number, cont.			

The Malware Event Data Block for 5.3+ Fields table describes the fields in the malware event data block.

Malware Event Data Block for 5.3+ Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always 35.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the FireAMP agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.

Malware Event Data Block for 5.3+ Fields (Continued)

Field	Data Type	Description
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint32	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Sourcefire Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.

Malware Event Data Block for 5.3+ Fields (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint8	The file type of the detected or quarantined file. The meaning of this field is transmitted in the metadata with this event. See FireAMP File Type Metadata on page 105 for more information.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.

Malware Event Data Block for 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always 0.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.

Malware Event Data Block for 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — UNKNOWN — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — UNAVAILABLE — The software was unable to send a request to the Sourcefire cloud for a disposition, or the Sourcefire cloud services did not respond to the request. • 5 — CUSTOM SIGNATURE — The file matches a user-defined hash, and is treated in a fashion designated by the user.

Malware Event Data Block for 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist

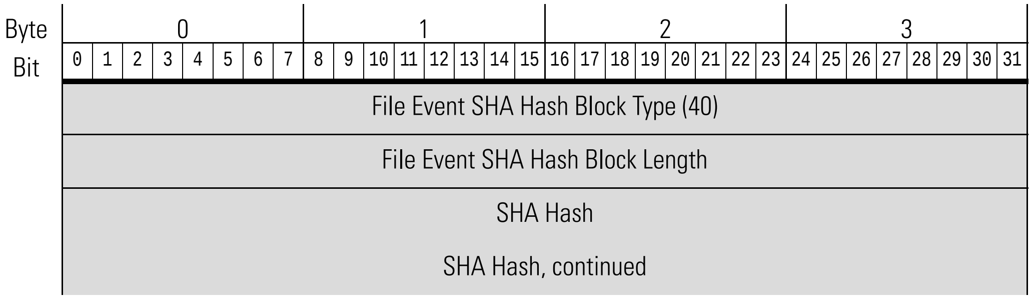
Malware Event Data Block for 5.3+ Fields (Continued)

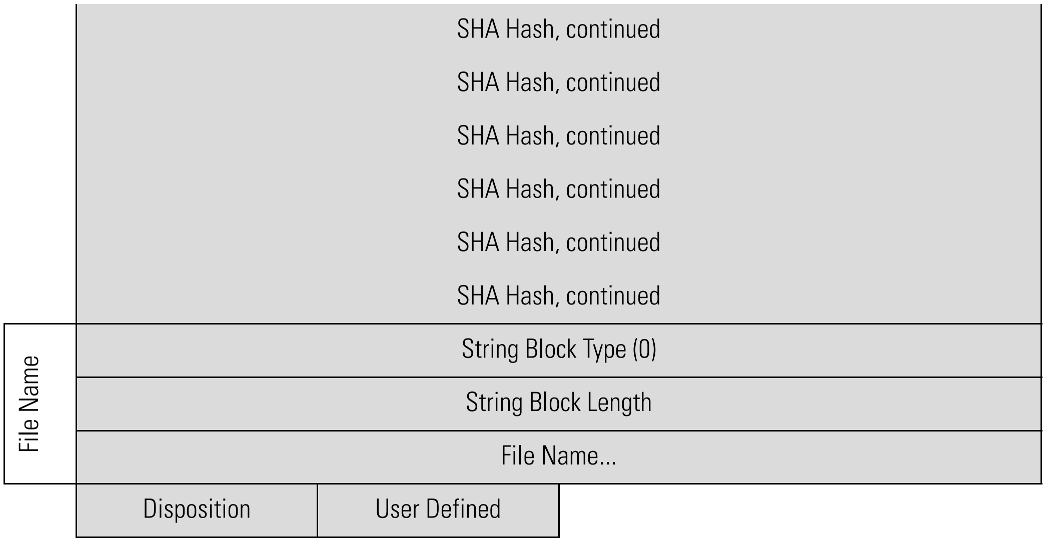
FIELD	DATA TYPE	DESCRIPTION
Protocol	uint8	IANA protocol number specified by the user. For example: • `1` — ICMP • `4` — IP • `6` — TCP • `17` — UDP This is currently only TCP.
Threat Score	uint8	A numeric value from 0 to 100 based on the potentially malicious behaviors observed during dynamic analysis.
IOC Number	uint16	ID Number of the compromise associated with this event.

File Event SHA Hash for 5.3+

The eStreamer service uses the File Event SHA Hash data block to contain metadata of the mapping of the SHA hash of a file to its filename. The block type is 40 in the series 2 list of data blocks. It can be requested if file log events have been requested in the extended requests—event code 111—and either bit 20 is set or metadata is requested with an event version of 5 and an event code of 21.

The following diagram shows the structure of a file event hash data block:





SHA Hash, continued

SHA Hash, continued

SHA Hash, continued

SHA Hash, continued

SHA Hash, continued

SHA Hash, continued

File Name

String Block Type (0)

String Block Length

File Name...

Disposition

User Defined

The File Event SHA Hash Data Block Fields table describes the fields in the file event SHA hash data block.

File Event SHA Hash Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
File Event SHA Hash Block Type	uint32	Initiates a File Event SHA Hash block. This value is always 26.
File Event SHA Hash Block Length	uint32	Total number of bytes in the File Event SHA Hash block, including eight bytes for the File Event SHA Hash block type and length fields, plus the number of bytes of data that follows.
SHA Hash	uint8[32]	The SHA-256 hash of the file in binary format.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the file. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.

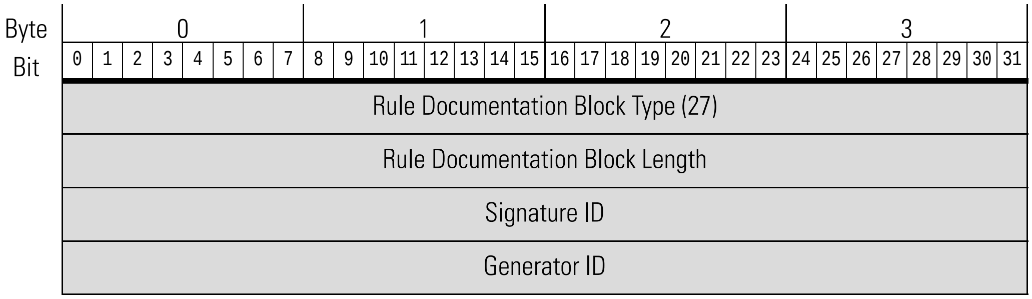
File Event SHA Hash Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
File Name or Disposition	string	The descriptive name or disposition of the file. If the file is clean, this value is `Clean`. If the file's disposition is unknown, the value is `Neutral`. If the file contains malware, the file name is given.
Disposition	uint8	The malware status of the file. Possible values include: • `1` — CLEAN — The file is clean and does not contain malware. • `2` — UNKNOWN — It is unknown whether the file contains malware. • `3` — MALWARE — The file contains malware. • `4` — UNAVAILABLE — The software was unable to send a request to the Sourcefire cloud for a disposition, or the Sourcefire cloud services did not respond to the request. • `5` — CUSTOM SIGNATURE — The file matches a user-defined hash, and is treated in a fashion designated by the user
User Defined	uint8	Indicated how the file name was provided: • `0` — defined by AMP • `1` — user defined

Rule Documentation Data Block for 5.2+

The eStreamer service uses the Rule Documentation data block to contain information about rules used to generate alerts. The block type is 27. It can be requested with a host request message of type 10. See Host Request Message Format on page 47 for more information.

The following diagram shows the structure of a rule documentation data block:



	Revision
Summary	String Block Type (0)
	String Block Length
	Summary
Impact	String Block Type (0)
	String Block Length
	Impact
Detailed Info	String Block Type (0)
	String Block Length
	Detailed Information
Affected Systems	String Block Type (0)
	String Block Length
	Affected Systems
Attack Scenarios	String Block Type (0)
	String Block Length
	Attack Scenarios
Ease of Attack	String Block Type (0)
	String Block Length
	Ease of Attack
False Positives	String Block Type (0)
	String Block Length
	False Positives
False Negatives	String Block Type (0)
	String Block Length
	False Negatives

Corrective Action	String Block Type (0)
	String Block Length
	Corrective Action
Contributors	String Block Type (0)
	String Block Length
	Contributors
Additional References	String Block Type (0)
	String Block Length
	Additional References

The Rule Documentation Data Block Fields table describes the fields in the rule documentation data block.

Rule Documentation Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Rule Documentation Data Block Type	uint32	Initiates a Rule Documentation data block. This value is always 27.
Rule Documentation Data Block Length	uint32	Total number of bytes in the Rule Documentation data block, including eight bytes for the Rule Documentation data block type and length fields, plus the number of bytes of data that follows.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
String Block Type	uint32	Initiates a String data block containing the summary associated with the rule. This value is always 0.

Rule Documentation Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Summary field.
Summary	string	Explanation of the threat or vulnerability.
String Block Type	uint32	Initiates a String data block containing the impact associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Impact field.
Impact	string	How a compromise that uses this vulnerability may impact various systems.
String Block Type	uint32	Initiates a String data block containing the detailed information associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detailed Information field.
Detailed Information	string	Information regarding the underlying vulnerability, what the rule actually looks for, and what systems are affected.
String Block Type	uint32	Initiates a String data block containing the list of affected systems associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Affected Systems field.
Affected Systems	string	Systems affected by the vulnerability.
String Block Type	uint32	Initiates a String data block containing the possible attack scenarios associated with the rule. This value is always 0.

Rule Documentation Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Attack Scenarios field.
Attack Scenarios	string	Examples of possible attacks.
String Block Type	uint32	Initiates a String data block containing the ease of attack associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Ease of Attack field.
Ease of Attack	string	Whether the attack is considered simple, medium, hard, or difficult, and whether or not is can be performed using a script.
String Block Type	uint32	Initiates a String data block containing the possible false positives associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the False Positives field.
False Positives	string	Examples that may result in a false positive. The default value is `None Known`.
String Block Type	uint32	Initiates a String data block containing the possible false negatives associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the False Negatives field.
False Negatives	string	Examples that may result in a false negative. The default value is `None Known`.
String Block Type	uint32	Initiates a String data block containing the corrective action associated with the rule. This value is always 0.

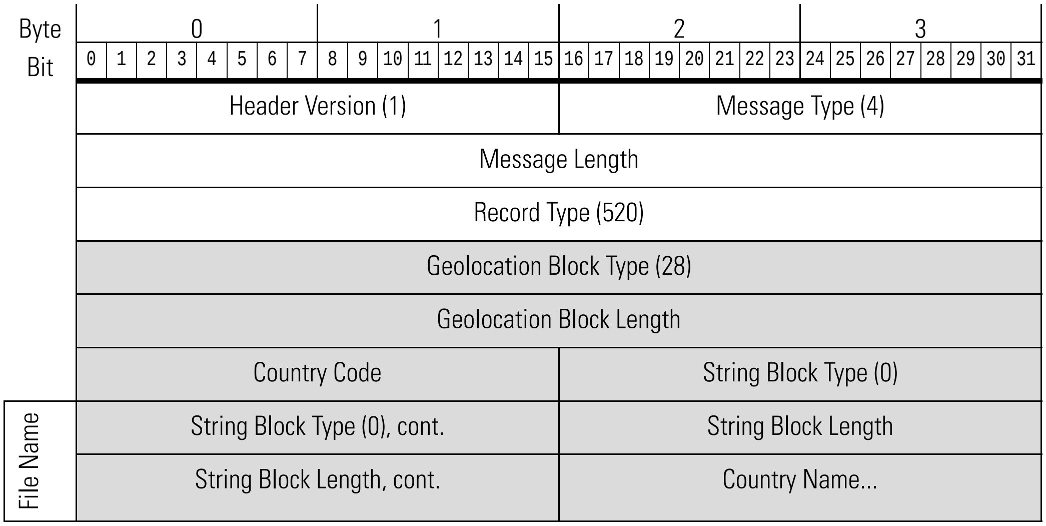
Rule Documentation Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Corrective Action field.
Corrective Action	string	Information regarding patches, upgrades, or other means to remove or mitigate the vulnerability.
String Block Type	uint32	Initiates a String data block containing the contributors for the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Contributors field.
Contributors	string	Contact information for the author of the rule and other relevant documentation.
String Block Type	uint32	Initiates a String data block containing the additional references associated with the rule. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Additional References field.
Additional References	string	Additional information and references.

Geolocation Data Block for 5.2+

This is a data block that contains the mapping of a country code to a country name. The record type is 520, and a block type of 28 in series 2. It is exposed as metadata for any event that has geolocation information. If metadata is requested and there is a value for the country code(s) in the event, then this block is returned along with other metadata.

The following diagram shows the structure of a geolocation data block:



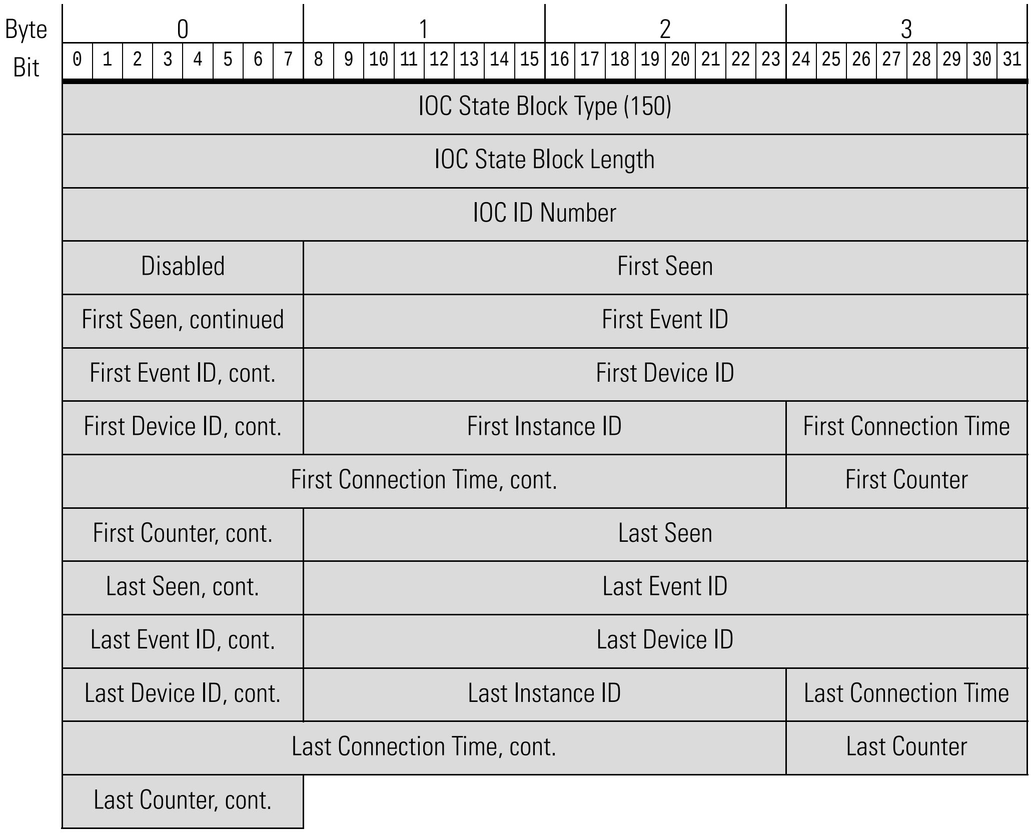
The Geolocation Data Block Fields table describes the fields in the Geolocation data block.

Geolocation Data Block Fields

Field	Data Type	Description
Geolocation Data Block Type	uint32	Initiates a Geolocation data block. This value is always 28.
Geolocation Data Block Length	uint32	Total number of bytes in the Geolocation data block, including eight bytes for the Geolocation data block type and length fields, plus the number of bytes of data that follows.
Country Code	uint16	The country code.
String Block Type	uint32	Initiates a String data block containing the country name associated with the country code. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Country Name field.
Country Name	string	The name of the country associated with the country code.

IOC State Data Block for 5.3+

The IOC State data block provides information about an Indication of Compromise (IOC). It is block type of 150 in series 1. It is used by the host tracker to store information about a compromise on a host. The following diagram shows the structure of an IOC State data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	IOC State Block Type (150)			
	IOC State Block Length			
	IOC ID Number			
	Disabled	First Seen		
	First Seen, continued	First Event ID		
	First Event ID, cont.	First Device ID		
	First Device ID, cont.	First Instance ID		First Connection Time
	First Connection Time, cont.			First Counter
	First Counter, cont.	Last Seen		
	Last Seen, cont.	Last Event ID		
	Last Event ID, cont.	Last Device ID		
	Last Device ID, cont.	Last Instance ID		Last Connection Time
	Last Connection Time, cont.			Last Counter
	Last Counter, cont.			

The IOC State Data Block Fields table describes the components of the IOC State data block.

.

IOC State Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
IOC State Data Block Type	uint32	Initiates an IOC State data block. This value is always 150.
IOC State Data Block Length	uint32	Total number of bytes in the IOC State data block, including eight bytes for the IOC State data block type and length fields, plus the number of bytes of data that follows.
IOC ID Number	uint32	Unique ID number for the compromise.
Disabled	uint8	Indicates whether the compromise has been disabled on the host: • `0` — The compromise is not disabled. • `1` — The compromise is disabled.
First Seen	uint32	Unix timestamp of when this compromise was first seen.
First Event ID	uint32	ID number of the event on which this compromise was first seen.
First Device ID	uint32	ID of the sensor which first detected the IOC.
First Instance ID	uint16	Numerical ID of the Snort instance on the managed device that first detected the compromise.
First Connection Time	uint32	Unix timestamp of the connection where this compromise was first seen.
First Counter	uint16	Counter for the connection on which this compromise was last seen. Used to differentiate between multiple connections occurring at the same time.
Last Seen	uint32	Unix timestamp of when this compromise was last seen
Last Event ID	uint32	ID number of the event on which this compromise was last seen.

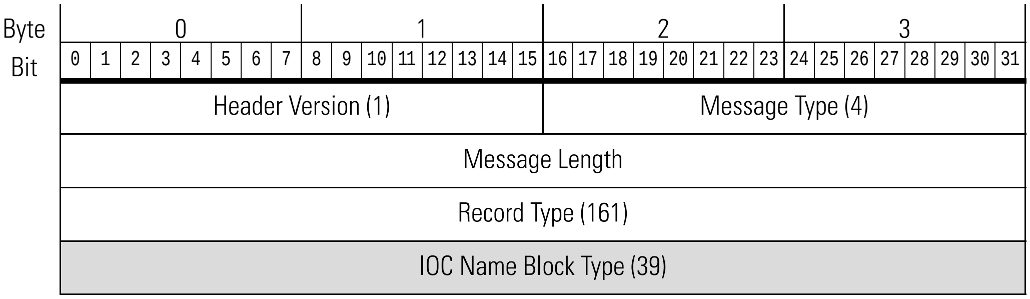
IOC State Data Block Fields (Continued)

Field	Data Type	Description
Last Device ID	uint32	ID of the sensor which most recently detected the IOC.
Last Instance ID	uint16	Numerical ID of the Snort instance on the managed device that last detected the compromise.
Last Connection Time	uint32	Unix timestamp of the connection on which this compromise was last seen.
Last Counter	uint16	Counter for the connection on which this compromise was last seen. Used to differentiate between multiple connections occurring at the same time.

IOC Name Data Block for 5.3+

This is a data block that provides the category and event type for an Indication of Compromise (IOC). The record type is 161, with a block type of 39 in series 2. It is exposed as metadata for any event that has IOC information. These include malware events, file events, and intrusion events.

The following diagram shows the structure of an IOC Name data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (161)			
	IOC Name Block Type (39)			

<table>
<tr><td></td><td colspan="2">IOC Name Block Length</td></tr>
<tr><td></td><td colspan="2">IOC ID Number</td></tr>
<tr><td rowspan="2">Category</td><td>String Block Type (0), cont.</td><td>String Block Length</td></tr>
<tr><td>String Block Length, cont.</td><td>Category...</td></tr>
<tr><td rowspan="2">Event Type</td><td>String Block Type (0), cont.</td><td>String Block Length</td></tr>
<tr><td>String Block Length, cont.</td><td>Event Type...</td></tr>
</table>

The IOC Name Data Block Fields table describes the fields in the IOC Name data block.

IOC Name Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
IOC Name Data Block Type	uint32	Initiates an IOC Name data block. This value is always 39.
IOC Name Data Block Length	uint32	Total number of bytes in the IOC Name data block, including eight bytes for the IOC Name data block type and length fields, plus the number of bytes of data that follows.
IOC ID Number	uint32	Unique ID number for the compromise.
String Block Type	uint32	Initiates a String data block containing the category associated with the compromise. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Category field.

IOC Name Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Category	string	The category for the compromise. Possible values include: • CnC Connected • Exploit Kit • High Impact Attack • Low Impact Attack • Malware Detected • Malware Executed • Dropper Infection • Java Compromise • Word Compromise • Adobe Reader Compromise • Excel Compromise • PowerPoint Compromise • QuickTime Compromise
String Block Type	uint32	Initiates a String data block containing the event type associated with the compromise. This value is always 0.

IOC Name Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Type field.
Event Type	string	The event type for the compromise. Possible values include: • Adobe Reader launched shell • Dropper Infection Detected by FireAMP • Excel Compromise Detected by FireAMP • Excel launched shell • Impact 1 Intrusion Event—attempted-admin • Impact 1 Intrusion Event—attempted-user • Impact 1 Intrusion Event—successful-admin • Impact 1 Intrusion Event—successful-user • Impact 1 Intrusion Event—web-application-attack • Impact 2 Intrusion Event—attempted-admin • Impact 2 Intrusion Event—attempted-user • Impact 2 Intrusion Event—successful-admin • Impact 2 Intrusion Event—successful-user • Impact 2 Intrusion Event—web-application-attack • Intrusion Event—exploit-kit • Intrusion Event—malware-backdoor • Intrusion Event—malware-CnC • Java Compromise Detected by FireAMP • Java launched shell • PDF Compromise Detected by FireAMP • PowerPoint Compromise Detected by FireAMP • PowerPoint launched shell • QuickTime Compromise Detected by FireAMP • QuickTime launched shell • Security Intelligence Event—CnC • Suspected Botnet Detected by FireAMP • Threat Detected by FireAMP—Subtype is 'executed' • Threat Detected by FireAMP—Subtype is not 'executed' • Threat Detected in File Transfer—Action is not 'block' • Word Compromise Detected by FireAMP • Word launched shell

CHAPTER 4

UNDERSTANDING DISCOVERY & CONNECTION DATA STRUCTURES

This chapter provides details about the data structures used in eStreamer messages for discovery and connection events, as well as the metadata for those events. Discovery and connection event messages use the same general message format and series of data blocks; the differences are in the contents of data blocks themselves.

Discovery events include two sub-categories of events:

- Host discovery events, which identify new and changed hosts on your managed network, including the applications running on the hosts detected from the contents of the packets, and the host vulnerabilities.
- User events, which report the detection of new users and user activity, such as logins.

Connection events report information about the session traffic between your monitored hosts and all other hosts. Connection information includes the first and last packet of the transaction, source and destination IP address, source and destination port, and the number of packets and bytes sent and received. If applicable, connection events also report the client application and URL involved in the session.

For information about requesting discovery or connection events from the eStreamer server, see Request Flags on page 30.

For information about the general structure of eStreamer event data messages, see Understanding the Organization of Event Data Messages on page 38.

See the following sections in this chapter for more information about discovery and connection event data structures:

- Discovery and Connection Event Data Messages on page 165 provides a high-level view of the structure that eStreamer uses for host discovery, user, and connection messages.
- Discovery and Connection Event Record Types on page 166 describes the record types for discovery and connection events.
- Metadata for Discovery Events on page 172 describes the metadata records that you can request for context information to convert numeric and coded data to text; for example, convert the user ID in an event to a user name.
- Discovery Event Header 5.2+ on page 198 describes the structure of the standard event header used in all discovery and connection messages, and the values that can occur in the event type and event subtype fields. The event type and subtype fields further define the structure of the data record carried in the message.
- Host Discovery Structures by Event Type on page 205 describes the structure of the data record that eStreamer uses for the various host discovery event types.
- User Data Structures by Event Type on page 222 describes the structure of the data record that eStreamer uses for the various user event types.
- Understanding Discovery (Series 1) Blocks on page 224 describes the series of data block structures that are used to convey complex records in discovery and connection event messages. Series 1 data blocks also appear in correlation events.
- User Vulnerability Data Block 5.0+ on page 336 describes other series 1 block structures that are used to convey complex user event records.

TIP! See "Appendix A: Data Structure Examples" on page 425 for examples that illustrate sample discovery events.

Discovery and Connection Event Data Messages

eStreamer packages the data for discovery and connection events in the same message structure, which contains:

- a record header that defines the record type
- a discovery event header that identifies and characterizes the event, and specifically identifies the event type and subtype. For information, see Discovery Event Header 5.2+ on page 198.
- a data record consisting of a block header and a data block. Discovery and connection event data messages use series 1 data blocks. For information, see Host Discovery and Connection Data Blocks on page 225 or User Vulnerability Data Block 5.0+ on page 336.

Discovery and Connection Event Record Types

The Discovery and Connection Event Record Types table below lists the event record types for host discovery and connection events, and provides links to the event message structure for each record type. The list includes metadata record types as well. Some records contain a single data block which stores a specific piece of data. These data blocks are broken up into series 1 blocks that contain most types of data, and series 2 blocks that specifically contain discovery data. The table also indicates the status of each version (current or legacy). A current record is the latest version. A legacy record has been superseded by a later version but can still be requested from eStreamer.

Discovery and Connection Event Record Types

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
10	139	1	New Host Detected	Current	New Host and Host Last Seen Messages on page 206
11	103	1	New TCP Server	Current	Server Messages on page 206
12	103	1	New UDP Server	Current	Server Messages on page 206
13	4	1	New Network Protocol	Current	New Network Protocol Message on page 207
14	4	1	New Transport Protocol	Current	New Transport Protocol Message on page 208
15	122	1	New Client Application	Current	Client Application Messages on page 208
16	103	1	TCP Server Information Update	Current	Server Messages on page 206
17	103	1	UDP Server Information Update	Current	Server Messages on page 206
18	53	1	OS Information Update	Current	Operating System Update Messages on page 210
19	N/A	N/A	Host Timeout	Current	IP Address Reused and Host Timeout/Deleted Messages on page 210
20	N/A	N/A	Host IP Address Reused	Current	IP Address Reused and Host Timeout/Deleted Messages on page 210

Discovery and Connection Event Record Types (Continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
21	N/A	N/A	Host Deleted: Host Limit Reached	Current	IP Address Reused and Host Timeout/Deleted Messages on page 210
22	N/A	N/A	Hops Change	Current	Hops Change Message on page 211
23	N/A	N/A	TCP Port Closed	Current	TCP and UDP Port Closed/Timeout Messages on page 212
24	N/A	N/A	UDP Port Closed	Current	TCP and UDP Port Closed/Timeout Messages on page 212
25	N/A	N/A	TCP Port Timeout	Current	TCP and UDP Port Closed/Timeout Messages on page 212
26	N/A	N/A	UDP Port Timeout	Current	TCP and UDP Port Closed/Timeout Messages on page 212
27	N/A	N/A	MAC Information Change	Current	MAC Address Messages on page 212
28	N/A	N/A	Additional MAC Detected for Host	Current	MAC Address Messages on page 212
29	N/A	N/A	Host IP Address Changed	Current	IP Address Change Message on page 209
30	139	1	Host Last Seen	Current	New Host and Host Last Seen Messages on page 206
31	N/A	N/A	Host Identified as Router/Bridge	Current	Host Identified as a Bridge/Router Message on page 213
32	8	1	Vulnerability Change	Current	Vulnerability Change Message on page 211
33	144	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.3+ on page 300

Discovery and Connection Event Record Types (Continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
33	152	1	Connection Statistics	Legacy	Connection Statistics Data Block 5.3+ on page 300
34	14	1	VLAN Tag Information Update	Current	VLAN Tag Information Update Messages on page 213
35	122	1	Client Application Timeout	Current	Client Application Messages on page 208
42	35	1	NetBIOS Name Change	Current	Change NetBIOS Name Message on page 213
44	N/A	N/A	Host Dropped: Host Limit Reached	Current	IP Address Reused and Host Timeout/Deleted Messages on page 210
45	37	1	Update Banner	Current	Update Banner Message on page 214
46	55	1	Add Host Attribute	Current	Attribute Messages on page 218
47	55	1	Update Host Attribute	Current	Attribute Messages on page 218
48	55	1	Delete Host Attribute	Current	Attribute Messages on page 218
51	103	1	TCP Server Confidence Update	Legacy	Server Messages on page 206
52	103	1	UDP Server Confidence Update	Legacy	Server Messages on page 206
53	53	1	OS Confidence Update	Legacy	Operating System Update Messages on page 210
54	N/A	N/A	Fingerprint Metadata	Current	Fingerprint Record on page 173
55	N/A	N/A	Client Application Metadata	Current	Client Application Record on page 174
57	N/A	N/A	Vulnerability Metadata	Current	Vulnerability Record on page 175
58	N/A	N/A	Criticality Metadata	Current	Criticality Record on page 178

Discovery and Connection Event Record Types (Continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
59	N/A	N/A	Network Protocol Metadata	Current	Network Protocol Record on page 179
60	N/A	N/A	Attribute Metadata	Current	Attribute Record on page 180
61	N/A	N/A	Scan Type Metadata	Current	Scan Type Record on page 181
63	N/A	N/A	Server Metadata	Current	Server Record on page 182
71	144	1	Connection Statistics	Current	Connection Statistics Data Block 5.3+ on page 300
73	136	1	Connection Chunks	Current	Connection Chunk Message on page 216
74	N/A	N/A	User Set OS	Current	User Server and Operating System Messages on page 219
75	N/A	N/A	User Set Server	Current	User Server and Operating System Messages on page 219
76	83	1	User Delete Protocol	Current	User Protocol Messages on page 220
77	60	1	User Delete Client Application	Current	User Client Application Messages on page 220
78	78	1	User Delete Address	Current	User Add and Delete Host Messages on page 217
79	77	1	User Delete Server	Current	User Delete Server Message on page 217
80	80	1	User Set Valid Vulnerabilities	Current	User Set Vulnerabilities Messages for Version 4.6.1+ on page 216
81	80	1	User Set Invalid Vulnerabilities	Current	User Set Vulnerabilities Messages for Version 4.6.1+ on page 216
82	81	1	User Set Host Criticality	Current	User Set Host Criticality Messages on page 218
83	55	1	User Set Attribute Value	Current	Attribute Value Messages on page 219

Discovery and Connection Event Record Types (Continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
84	82	1	User Delete Attribute Value	Current	Attribute Value Messages on page 219
85	78	1	User Add Host	Current	User Add and Delete Host Messages on page 217
86	N/A	N/A	User Add Server	Current	User Server and Operating System Messages on page 219
87	60	1	User Add Client Application	Current	User Client Application Messages on page 220
88	83	1	User Add Protocol	Current	User Protocol Messages on page 220
89	142	1	User Add Scan Result	Current	Add Scan Result Messages on page 221
90	N/A	N/A	Source Type Record	Current	Source Type Record on page 183
91	N/A	N/A	Source Application Record	Current	Source Application Record on page 184
92	120	1	User Dropped Change Event	Current	User Modification Messages on page 223
93	120	1	User Removed Change Event	Current	User Modification Messages on page 223
94	120	1	New User Identification Event	Current	User Modification Messages on page 223
95	121	1	User Login Change Event	Current	User Information Update Message Block on page 223
96	N/A	N/A	Source Detector Record	Current	Source Detector Record on page 185
98	N/A	N/A	User Record	Current	User Record on page 188
101	N/A	N/A	New OS Event	Current	New Operating System Messages on page 221

Discovery and Connection Event Record Types (Continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
102	94	1	Identity Conflict Event	Current	Identity Conflict and Identity Timeout System Messages on page 222
103	94	1	Identity Timeout Event	Current	Identity Conflict and Identity Timeout System Messages on page 222
106	N/A	N/A	Third Party Scanner Vulnerability Record	Current	Third Party Scanner Vulnerability Record on page 186
107	122	1	Client Application Update	Current	Client Application Messages on page 208
109	N/A	N/A	Web Application Record	Current	Web Application Record on page 189
115	N/A	N/A	Security Zone Name Record	Current	Security Zone Name Record on page 93
116	14	2	Interface Name Record	Current	Interface Name Record on page 94
117	114	1	Access Control Policy Name Metadata	Current	Access Control Policy Name Record on page 96
118	14	2	Intrusion Policy Name Record	Current	Intrusion Policy Name Record on page 190
119	14	2	Access Control Rule ID Record	Current	Access Control Rule ID Record Metadata on page 97
120	N/A	N/A	Access Control Rule Action Record	Current	Access Control Rule Action Record Metadata on page 191
121	N/A	N/A	URL Category Record	Current	URL Category Record Metadata on page 192
122	N/A	N/A	URL Reputation Metadata	Current	URL Reputation Record Metadata on page 193

Discovery and Connection Event Record Types (Continued)

Record Type	Contains Block Type	Series	Description	Record Status	Data Format Described in...
124	21	2	Access Control Rule Reason Metadata	Current	Access Control Rule Reason Metadata on page 194
280	22	2	Security Intelligence Category Metadata	Current	Security Intelligence Category Metadata on page 196
281	N/A	N/A	Security Intelligence Source/Destination Metadata	Current	Security Intelligence Source/Destination Record on page 197

Metadata for Discovery Events

You request metadata by metadata version number. For the metadata version that corresponds to your version of the Sourcefire 3D System, see Understanding Metadata on page 63. For important information on how eStreamer streams metadata records, see Metadata Transmission on page 63.

For information on the structures of the various metadata records types for host discovery and user event records, see:

- Fingerprint Record on page 173
- Client Application Record on page 174
- Vulnerability Record on page 175
- Criticality Record on page 178
- Network Protocol Record on page 179
- Attribute Record on page 180
- Scan Type Record on page 181
- Server Record on page 182
- Source Type Record on page 183
- Source Application Record on page 184
- Source Detector Record on page 185
- Third Party Scanner Vulnerability Record on page 186
- User Record on page 188
- Web Application Record on page 189
- Intrusion Policy Name Record on page 190

- Access Control Rule Action Record Metadata on page 191
- URL Category Record Metadata on page 192
- URL Reputation Record Metadata on page 193

For metadata records for intrusion and correlation events, see Intrusion Event and Metadata Record Types on page 64.

Fingerprint Record

The eStreamer service transmits the fingerprint metadata for an event within a Fingerprint record, the format of which is shown below. (Fingerprint metadata is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 54, indicating a Fingerprint record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (54)			
	Record Length			
Fingerprint UUID	Fingerprint UUID			
	Fingerprint UUID cont.			
	Fingerprint UUID cont.			
	Fingerprint UUID cont.			
	OS Name Length			
	OS Name...			
	OS Vendor Length			
	OS Vendor...			
	OS Version Length			
	OS Version...			

The Fingerprint Record Fields table describes the fields in the Fingerprint record.

Fingerprint Record Fields

FIELD	DATA TYPE	DESCRIPTION
Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the operating system.
OS Name Length	uint32	The number of bytes included in the operating system name.
OS Name	string	The name of the operating system for the fingerprint.
OS Vendor Length	uint32	The number of bytes included in the operating system vendor name.
OS Vendor	string	The name of the operating system vendor for the fingerprint.
OS Version Length	uint32	The number of bytes included in the operating system version.
OS Version	string	The version of the operating system for the fingerprint.

Client Application Record

The eStreamer service transmits the client application metadata for an event within a Client Application record, the format of which is shown below. (Client application metadata is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 55, indicating a Client Application record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (55)			
	Record Length			
	Application ID			
	Name Length			
	Name...			

The Client Application Record Fields table describes the fields in the Client Application record.

Client Application Record Fields

Field	Data Type	Description
Application ID	uint32	The application ID number for the client application.
Name Length	uint32	The number of bytes included in the name.
Name	string	The client application name.

Vulnerability Record

The eStreamer service transmits metadata containing vulnerability information for an event within a Vulnerability record, the format of which is shown below. (Vulnerability information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 57, indicating a Vulnerability record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (57)			
	Record Length			
	Vulnerability ID			
	Impact			
	Exploits	Remote	Entry Date Length	
	Entry Date Length Cont.		Entry Date...	
	Published Date Length			
	Published Date...			
	Modified Date Length			
	Modified Date...			
	Title Length			
	Title...			
	Short Description Length			
	Short Description...			
	Description Length			
	Description...			
	Technical Description Length			
	Technical Description...			
	Solution Length			
	Solution...			

The Vulnerability Record Fields table describes the fields in the Vulnerability record.

Vulnerability Record Fields

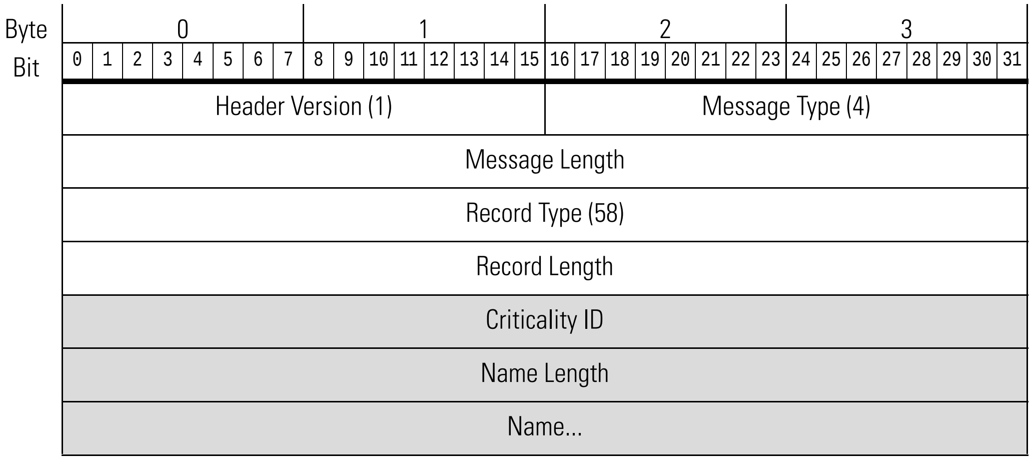
FIELD	DATA TYPE	DESCRIPTION
Vulnerability ID	uint32	The vulnerability ID number.
Impact	uint32	The vulnerability impact, corresponding to the impact level determined through correlation of intrusion data, host discovery events, and vulnerability assessments. The value can be from 1 to 10, with 10 being the most severe. The impact value of a vulnerability is determined by the writer of the Bugtraq entry.
Exploits	uint8	Indicates whether known exploits exist for the vulnerability. Possible values include: • `0` — Yes • `1` — No
Remote	uint8	Indicates whether the vulnerability can be exploited across a network. Possible values include: • `0` — Yes • `1` — No • Blank — Vulnerability to remote exploits unknown
Entry Date Length	uint32	The length of the entry date field.
Entry Date	string	The date the vulnerability was entered in the database.
Published Date Length	uint32	The length of the published date field.
Published Date	string	The date the vulnerability was published.
Modified Date Length	uint32	The length of the modified date field.
Modified Date	string	The date of the most recent modification to the vulnerability, if applicable.
Title Length	uint32	The length of the title field.

Vulnerability Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Title	string	The title of the vulnerability.
Short Description Length	uint32	The length of the short description field.
Short Description	string	A summary description of the vulnerability.
Description Length	uint32	The length of the description field.
Description	string	A general description of the vulnerability.
Technical Description Length	uint32	The length of the technical description field.
Technical Description	string	The technical description of the vulnerability.
Solution Length	uint32	The length of the solution field.
Solution	string	The solution to the vulnerability.

Criticality Record

The eStreamer service transmits metadata containing host criticality information for an event within a Criticality record, the format of which is shown below. (Criticality information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 58, indicating a Criticality record.



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Record Type (58)																															
	Record Length																															
	Criticality ID																															
	Name Length																															
	Name...																															

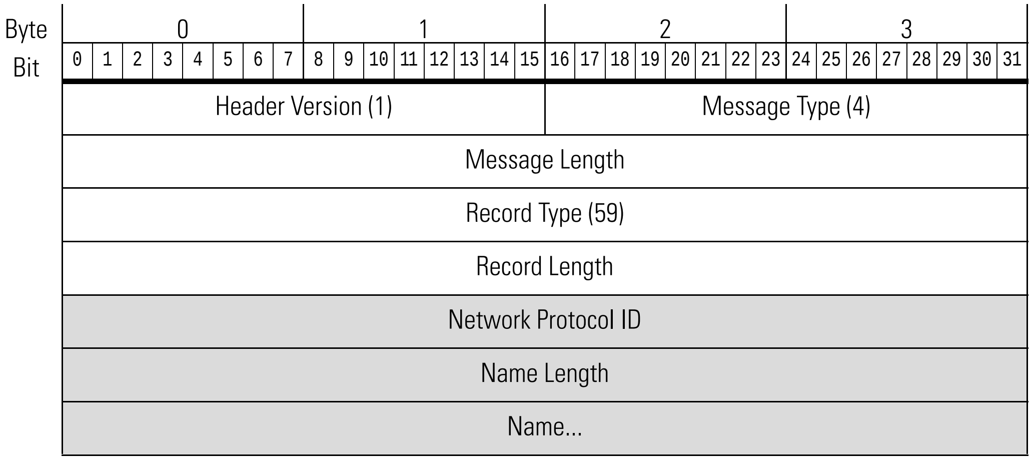
The Criticality Record Fields table describes the fields in the Criticality record.

Criticality Record Fields

FIELD	DATA TYPE	DESCRIPTION
Criticality ID	uint32	The criticality ID number.
Name Length	uint32	The number of bytes included in the criticality level.
Name	string	The criticality level.

Network Protocol Record

The eStreamer service transmits metadata containing network protocol information for an event within a Network Protocol record, the format of which is shown below. (Network protocol information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 59, indicating a Network Protocol record.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (59)			
	Record Length			
	Network Protocol ID			
	Name Length			
	Name...			

The Network Protocol Record Fields table describes the fields in the Network Protocol record.

Network Protocol Record Fields

Field	Data Type	Description
Network Protocol ID	uint32	The network protocol ID number.
Name Length	uint32	The number of bytes included in the network protocol name.
Name	string	The name of the network protocol.

Attribute Record

The eStreamer service transmits metadata containing attribute information for an event within an Attribute record, the format of which is shown below. (Attribute information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 60, indicating an Attribute record.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (60)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">Attribute ID</td></tr>
<tr><td></td><td colspan="32">Name Length</td></tr>
<tr><td></td><td colspan="32">Name...</td></tr>
</table>

The Attribute Record Fields table describes the fields in the Attribute record.

Attribute Record Fields

FIELD	DATA TYPE	DESCRIPTION
Attribute ID	uint32	The attribute ID number.
Name Length	uint32	The number of bytes included in the attribute name.
Name	string	The name of the attribute.

Scan Type Record

The eStreamer service transmits metadata containing scan type information for an event within a Scan Type record, the format of which is shown below. (Scan type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 61, indicating a Scan Type record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (61)			
	Record Length			
	Scan Type ID			
	Name Length			
	Name...			

The Scan Type Record Fields table describes the fields in the Scan Type record.

Scan Type Record Fields

Field	Data Type	Description
Scan Type ID	uint32	The scan type ID number.
Name Length	uint32	The number of bytes included in the scan type name.
Name	string	The name of the scan type.

Server Record

The eStreamer service transmits metadata containing server information for an event within a Server record, the format of which is shown below. The application ID of the server's application protocol provides the cross-reference to the metadata. (Server information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 63, indicating a Server record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (63)			
	Record Length			
	Application ID			
	Name Length			
	Name...			

The Server Record Fields table describes the fields in the Server record.

Server Record Fields

FIELD	DATA TYPE	DESCRIPTION
Application ID	uint32	The application ID number of the application protocol.
Name Length	uint32	The number of bytes included in the server name.
Name	string	The name of the application protocol. For application ID 65535, the name is `unknown`.

Source Type Record

The eStreamer service transmits metadata containing information about the source application for an event within a Source Type record, the format of which is shown below. (Source type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 90, indicating a Source Type record.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Record Type (90)																															
	Record Length																															
	Source Type ID																															
	Name Length																															
	Name...																															

The Source Type Record Fields table describes the fields in the Source Type record.

Source Type Record Fields

FIELD	DATA TYPE	DESCRIPTION
Source Type ID	uint32	The identification number for the source type.
Name Length	uint32	The number of bytes included in the source type name.
Name	string	The name of the source type.

Source Application Record

The eStreamer service transmits metadata containing information about the source application for a host discovery event within a Source Application record, the format of which is shown below. (Source application information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 91, indicating a Source Application record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (91)			
	Record Length			
	Source Application ID			
	Name Length			
	Name...			

The Source Application Record Fields table describes the fields in the Source Application record.

Source Application Record Fields

Field	Data Type	Description
Source Application ID	uint32	The ID number for the source application.
Name Length	uint32	The number of bytes included in the source application name.
Name	string	The name of the source application.

Source Detector Record

The eStreamer service transmits metadata containing information about the source application for a host discovery event within a Source Type record, the format of which is shown below. (Source type information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 96, indicating a Source Detector record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (96)			
	Record Length			
	Source Detector ID			
	Name Length			
	Name...			

The Source Detector Record Fields table describes the fields in the Source Detector record.

Source Detector Record Fields

FIELD	DATA TYPE	DESCRIPTION
Source Detector ID	uint32	The ID string for the source detector.
Name Length	uint32	The number of bytes included in the source type name.
Name	string	The name of the source detector.

Third Party Scanner Vulnerability Record

The eStreamer service transmits metadata containing third party vulnerability information for an event within a Third Party Scanner Vulnerability record, the format of which is shown below. (Vulnerability information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 106, indicating a Third Party Scanner Vulnerability record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (106)			
	Record Length			
	Vulnerability ID			
	Scanner Type			
	Title Length			
	Title...			
	Description Length			
	Description...			
	CVE ID Length			
	CVE ID...			
	BugTraq Length			
	BugTraq ID...			

The Third Party Scanner Vulnerability Record Fields table describes the fields in the Vulnerability record.

Third Party Scanner Vulnerability Record Fields

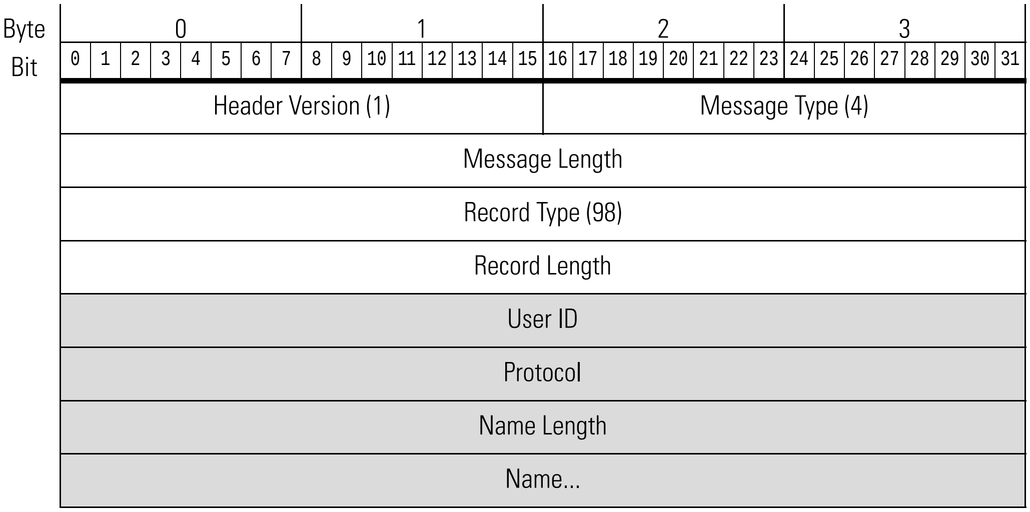
Field	Data Type	Description
Vulnerability ID	uint32	The third party vulnerability ID number.
Scanner Type	uint32	The third party scanner type.
Title Length	uint32	The length of the title field.
Title	string	The title of the vulnerability.
Description Length	uint32	The length of the description field.

Third Party Scanner Vulnerability Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Description	string	A general description of the vulnerability.
CVE ID Length	uint32	The length of the CVE ID field.
CVE ID	string	The Common Vulnerabilities and Exposures (CVE) ID number for the vulnerability.
BugTraq ID Length	uint32	The length of the BugTraq ID field.
BugTraq ID	string	The BugTraq ID number for the vulnerability.

User Record

The eStreamer service transmits metadata containing information about users detected by the system within a User record, the format of which is shown below. (User information is sent when the Version 4 metadata and the policy event request flag—bits 20 and 22, respectively, in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 98, indicating a User record.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (98)			
	Record Length			
	User ID			
	Protocol			
	Name Length			
	Name...			

The User Record Fields table describes the fields in the User record.

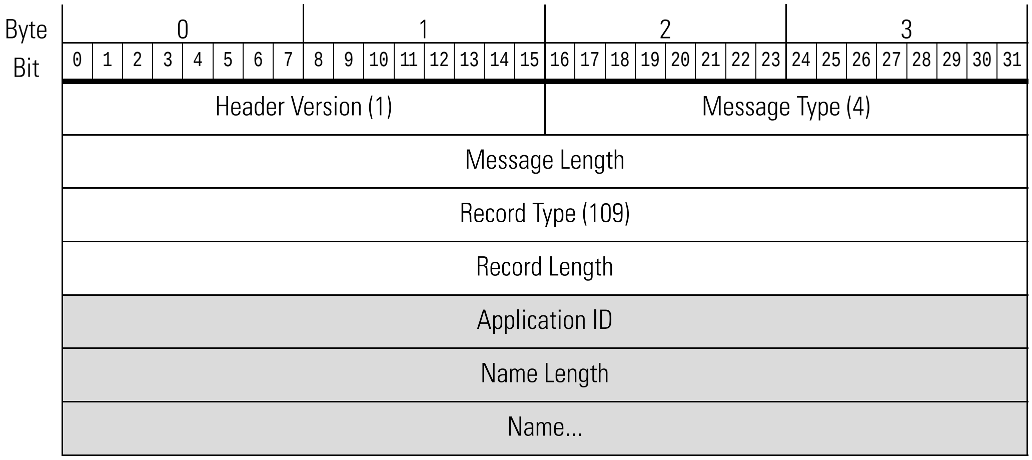
User Record Fields

FIELD	DATA TYPE	DESCRIPTION
User ID	uint32	The ID string for the user.
Protocol	uint32	The protocol for the traffic where the user was detected.
Name Length	uint32	The number of bytes included in the user name.
Name	string	The name of the user.

Web Application Record

The system detects the content of HTTP traffic from websites, if available. Web application metadata for a host discovery event may include the specific type of content (for example, WMV or QuickTime).

The eStreamer service transmits the web application metadata for an event within a Web Application record, the format of which is shown below. (Web application metadata is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 109, indicating a Web Application record.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (109)			
	Record Length			
	Application ID			
	Name Length			
	Name...			

The Web Application Record Fields table describes the fields in the Web Application record.

Web Application Record Fields

FIELD	DATA TYPE	DESCRIPTION
Application ID	uint32	Application ID number of the web application.
Name Length	uint32	The number of bytes included in the name.
Name	string	The web application content name.

Intrusion Policy Name Record

The eStreamer service transmits metadata containing intrusion policy name information for a connection event within an Intrusion Policy Name record, the format of which is shown below. (Intrusion policy name information is sent when one of the metadata flags—version 4 metadata bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Intrusion Policy Name record field, which appears after the Message Length field, has a value of 118, indicating an Intrusion Policy Name record.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type (118)</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">Intrusion Policy Name Data Block (14)</td></tr>
<tr><td></td><td colspan="32">Intrusion Policy Name Data Block Length</td></tr>
<tr><td></td><td colspan="32">Intrusion Policy UUID
Intrusion Policy UUID, continued
Intrusion Policy UUID, continued
Intrusion Policy UUID, continued</td></tr>
<tr><td></td><td colspan="32">String Block Type (0)</td></tr>
<tr><td></td><td colspan="32">String Block Length</td></tr>
<tr><td></td><td colspan="32">Intrusion Policy Name...</td></tr>
</table>

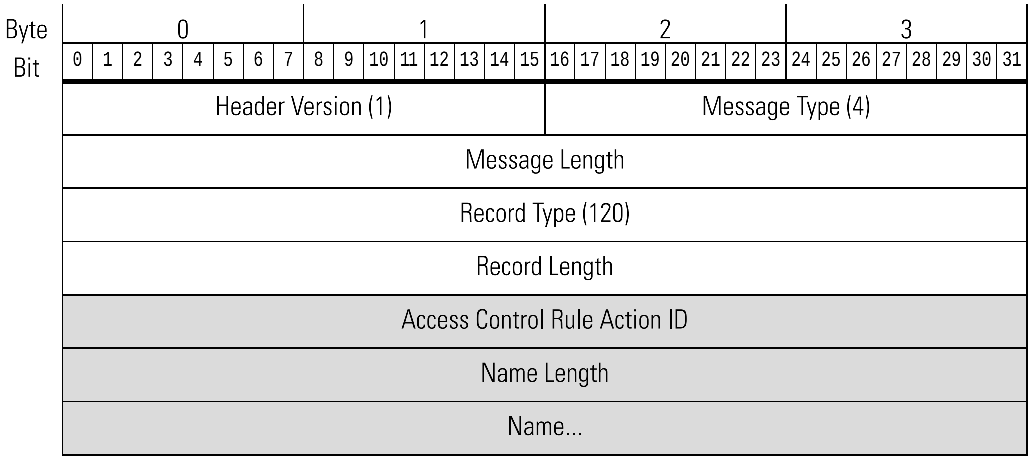
The Intrusion Policy Name Data Block Fields table describes the fields in the Intrusion Policy Name data block.

Intrusion Policy Name Data Block Fields

Field	Data Type	Description
Intrusion Policy Name Data Block Type	uint32	Initiates an Intrusion Policy Name data block. This value is always 14. The block type is a series 2 block.
Intrusion Policy Name Data Block Length	uint32	Length of the data block. Includes the number of bytes of data plus the 8 bytes in the two data block header fields.
Intrusion Policy UUID	uint8[16]	The unique identifier for the intrusion policy associated with the connection event.
String Block Type	uint32	Initiates a String data block containing the name of the intrusion policy. This value is always 0.
String Block Length	uint32	The number of bytes included in the intrusion policy name String data block, including eight bytes for the block type and header fields plus the number of bytes in the intrusion policy name.
Intrusion Policy Name	string	The intrusion policy name.

Access Control Rule Action Record Metadata

The eStreamer service transmits metadata containing the action associated with a triggered access control rule within an Access Control Rule Action record, the format of which is shown below. (Access Control Rule Action information is sent when the version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Access Control Rule Action record field, which appears after the Message Length field, has a value of 120, indicating an Access Control Rule Action record.



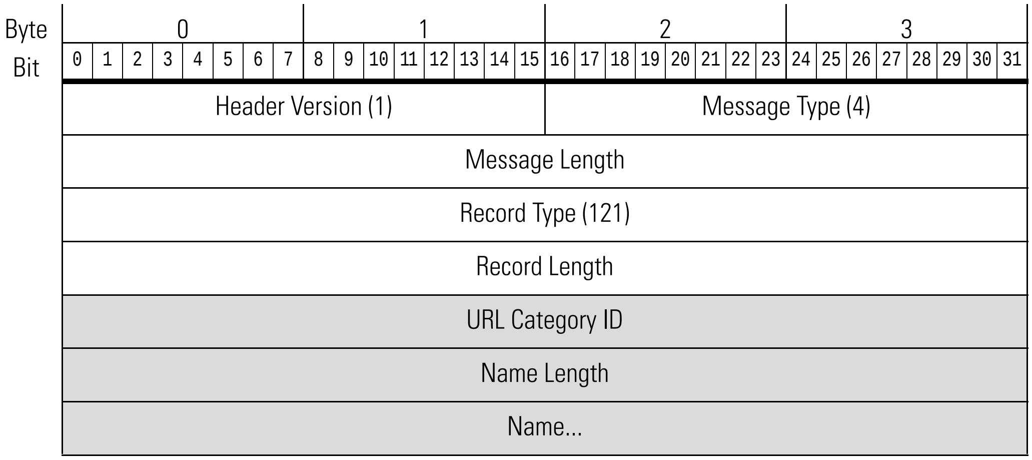
The Access Control Rule Action Record Fields table describes the fields in the Access Control Rule Action record.

Access Control Rule Action Record Fields

FIELD	DATA TYPE	DESCRIPTION
Access Control Rule Action ID	uint32	ID number of the access control rule action.
Name Length	uint32	The number of bytes included in the name.
Name	string	The firewall rule action name.

URL Category Record Metadata

The eStreamer service transmits metadata containing the category name associated with a URL in a connection log within a URL Category record, the format of which is shown below. (URL category information is sent when the version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the record field, which appears after the Message Length field, has a value of 121, indicating a URL Category record.



The URL Category Record Fields table describes the fields in the URL Category record.

URL Category Record Fields

FIELD	DATA TYPE	DESCRIPTION
URL Category ID	uint32	ID number of the URL category.
Name Length	uint32	The number of bytes included in the name.
Name	string	The URL category name.

URL Reputation Record Metadata

The eStreamer service transmits metadata containing the reputation (that is, risk level) associated with a URL in a connection log within a URL Reputation record, the format of which is shown below. (URL reputation information is sent when the version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the URL Reputation metadata record field, which appears after the Message Length field, has a value of 122, indicating a URL Reputation metadata record.

<table>
<tr><td>Byte</td><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td></tr>
<tr><td></td><td colspan="2">Header Version (1)</td><td colspan="2">Message Type (4)</td></tr>
<tr><td></td><td colspan="4">Message Length</td></tr>
<tr><td></td><td colspan="4">Record Type (122)</td></tr>
<tr><td></td><td colspan="4">Record Length</td></tr>
<tr><td></td><td colspan="4">URL Reputation ID</td></tr>
<tr><td></td><td colspan="4">Name Length</td></tr>
<tr><td></td><td colspan="4">Name...</td></tr>
</table>

The URL Reputation Record Fields table describes the fields in the URL Reputation record.

URL Reputation Record Fields

FIELD	DATA TYPE	DESCRIPTION
URL Reputation ID	uint32	ID number of the URL reputation.
Name Length	uint32	The number of bytes included in the name.
Name	string	The URL reputation name.

Access Control Rule Reason Metadata

The eStreamer service transmits metadata containing information about the reason an access control rule triggered an intrusion event or connection event within an Access Control Rule Reason record, the format of which is shown below. Access control rule reason metadata is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30. Note that the Record Type field, which appears after the Message Length field, has a value of 124, indicating an Access Control Rule Reason record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (124)			
	Record Length			
	Access Control Rule Reason Block Type (21)			
	Access Control Rule Block Length			
	Access Control Rule Reason		String Block Type (0)	
	String Block Type (0), cont.		String Block Length	
	String Block Length, cont.		Description...	

The Access Control Rule Reason Metadata Fields table describes the fields in the Access Control Rule ID data block.

Access Control Rule Reason Metadata Fields

FIELD	DATA TYPE	DESCRIPTION
Access Control Rule Reason Block Type	uint32	Initiates an Access Control Rule Reason block. This value is always 21. This is a series 2 data block.
Access Control Rule Reason Block Length	uint32	Total number of bytes in the Access Control Rule Reason block, including eight bytes for the Access Control Rule Reason block type and length fields, plus the number of bytes of data that follows.
Access Control Rule Reason	uint16	The reason the Access Control rule logged the connection.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the Access Control rule reason.

Security Intelligence Category Metadata

The eStreamer service transmits metadata containing information about the Security Intelligence category within a Security Intelligence Category record, the format of which is shown below. Access control rule reason metadata is sent when the Version 4 metadata flag—bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30. Note that the Record Type field, which appears after the Message Length field, has a value of 280, indicating a Security Intelligence Category record.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (280)			
	Record Length			
	Security Intelligence Category Block Type (22)			
	Security Intelligence Category Block Length			
	Security Intelligence List ID			
	Access Control Policy UUID			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	String Block Type (0)			
	String Block Length			
	Security Intelligence List Name...			

The Security Intelligence Category Metadata Fields table describes the fields in the Security Intelligence Category record.

Security Intelligence Category Metadata Fields

FIELD	DATA TYPE	DESCRIPTION
Security Intelligence Category Block Type	uint32	Initiates an Security Intelligence Category data block. This value is always 22. This is a series 2 data block.
Security Intelligence Category Block Length	uint32	Total number of bytes in the Security Intelligence Category block, including eight bytes for the Security Intelligence Category block type and length fields, plus the number of bytes of data that follows.
Security Intelligence List ID	uint32	The ID of the IP blacklist or whitelist triggered by the connection.
Access Control Policy UUID	uint8[16]	The UUID of the access control policy configured for Security Intelligence.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Security Intelligence List Name field.
Security Intelligence List Name	string	The name of the IP category blacklist or whitelist triggered by the connection.

Security Intelligence Source/Destination Record

The eStreamer service transmits metadata containing whether a Security Intelligence-detected IP address is a source IP address or destination IP address within a Security Intelligence Source/Destination record, the format of which is shown below. (The source/destination IP information is sent when one of the metadata flags—bits 1, 14, 15, or 20 in the Request Flags field of a request message—is set. See Request Flags on page 30.) Note that the Record Type field, which appears after the Message Length field, has a value of 281, indicating a Security Intelligence Source/Destination record.

Byte	0		1		2		3	
Bit	0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15	16 17 18 19	20 21 22 23	24 25 26 27	28 29 30 31
	Header Version (1)				Message Type (4)			
	Message Length							
	Record Type (281)							
	Record Length							
	Security Intelligence Source/Destination ID							
	Security Intelligence Source/Destination Length							
	Security Intelligence Source/Destination...							

The Security Intelligence Source/Destination Record Fields table describes the fields in the Security Intelligence Source/Destination record.

Security Intelligence Source/Destination Record Fields

FIELD	DATA TYPE	DESCRIPTION
Security Intelligence Source/ Destination ID	uint32	The Security Intelligence source/destination ID number.
Security Intelligence Source/ Destination Length	uint32	The number of bytes included in the Security Intelligence source/destination.
Security Intelligence Source/ Destination	string	Whether the detected IP address is a source or destination IP address.

Discovery Event Header 5.2+

Discovery and connection event messages contain a discovery event header. It conveys the type and subtype of the event, the time the event occurred, the device on which the event occurred, and the structure of the event data in the message. This header is followed by the actual host discovery, user, or connection event data. The structures associated with the different event type/subtype values are described in Host Discovery Structures by Event Type on page 205. This header has IPv6 support, and deprecates Discovery Event Header

4.8.0.2-5.1.1.x on page 514.

The event type and event subtype fields of the discovery event header identify the structure of the transmitted event message. Once the structure of the event data block is determined, your program can parse the message appropriately.

The shaded rows in the following diagram illustrate the format of the discovery event header.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
Discovery Event Header	Device ID			
	Legacy IP Address			
	MAC Address			
	MAC Address, continued		Has IPv6	Reserved for future use
	Event Second			
	Event Microsecond			
	Event Type			
	Event Subtype			
	File Number (Internal Use Only)			
	File Position (Internal Use Only)			
	IPv6 Address			
	IPv6 Address, continued			
	IPv6 Address, continued			
	IPv6 Address, continued			

The Discovery Event Header Fields table describes the discovery event header.

Discovery Event Header Fields

FIELD	DATA TYPES	DESCRIPTION
Device ID	uint32	ID number of the device that generated the discovery event. You can obtain the metadata for the device by requesting Version 3 and 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Legacy IP Address	uint32	IPv4 address of the host involved in the event. If the Has IPv6 flag is set, this will contain `0.0.0.0`.
MAC Address	uint8[6]	MAC address of the host involved in the event.
Has IPv6	uint8	Flag indicating that the host has an IPv6 address.
Reserved for future use	uint8	Reserved for future use
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) that the system generated the event.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the system generated the event.
Event Type	uint32	Event type (1000 for new events, 1001 for change events, 1002 for user input events, 1050 for full host profile). See Host Discovery Structures by Event Type on page 205 for a list of available event types.
Event Subtype	uint32	Event subtype. See Host Discovery Structures by Event Type on page 205 for a list of available event subtypes.
File Number	byte[4]	Serial file number. This field is for Sourcefire internal use and can be disregarded.
File Position	byte[4]	Event's position in the serial file. This field is for Sourcefire internal use and can be disregarded.
IPv6 Address	uin8[16]	IPv6 address. This field is present and used if the Has IPv6 flag is set.

Discovery and Connection Event Types and Subtypes

The values in the Event Type and Event Subtype fields identify and classify the event contained in a host discovery or user data message. They also identify the structure of the data in the message.

The Discovery and Connection Events by Type and Subtype table lists the event types and event subtypes for discovery and connection events.

Discovery and Connection Events by Type and Subtype

Event Name	Event Type	Event Subtype
New Host	1000	1
New TCP Server	1000	2
New Network Protocol	1000	3
New Transport Protocol	1000	4
New IP to IP Traffic	1000	5
New UDP Server	1000	6
New Client Application	1000	7
New OS	1000	8
New IPv6 to IPv6 Traffic	1000	9
Host IP Address Changed	1001	1
OS Information Update	1001	2
Host IP Address Reused	1001	3
Vulnerability Change	1001	4
Hops Change	1001	5
TCP Server Information Update	1001	6
Host Timeout	1001	7
TCP Port Closed	1001	8
UDP Port Closed	1001	9

Discovery and Connection Events by Type and Subtype (Continued)

EVENT NAME	EVENT TYPE	EVENT SUBTYPE
UDP Server Information Update	1001	10
TCP Port Timeout	1001	11
UDP Port Timeout	1001	12
MAC Information Change	1001	13
Additional MAC Detected for Host	1001	14
Host Last Seen	1001	15
Host Identified as Router/Bridge	1001	16
Connection Statistics	1001	17
VLAN Tag Information Update	1001	18
Host Deleted: Host Limit Reached	1001	19
Client Application Timeout	1001	20
NetBIOS Name Change	1001	21
NetBIOS Domain Change	1001	22
Host Dropped: Host Limit Reached	1001	23
Banner Update	1001	24
TCP Server Confidence Update	1001	25
UDP Server Confidence Update	1001	26
Identity Conflict	1001	29
Identity Timeout	1001	30
Secondary Host Update	1001	31
Client Application Update	1001	32
User Set Valid Vulnerabilities (Legacy)	1002	1

Discovery and Connection Events by Type and Subtype (Continued)

Event Name	Event Type	Event Subtype
User Set Invalid Vulnerabilities (Legacy)	1002	2
User Delete Address (Legacy)	1002	3
User Delete Server (Legacy)	1002	4
User Set Host Criticality	1002	5
Host Attribute Add	1002	6
Host Attribute Update	1002	7
Host Attribute Delete	1002	8
Host Attribute Set Value (Legacy)	1002	9
Host Attribute Delete Value (Legacy)	1002	10
Add Scan Result	1002	11
User Set Vulnerability Qualification	1002	12
User Policy Control	1002	13
Delete Protocol	1002	14
Delete Client Application	1002	15
User Set Operating System	1002	16
User Account Seen	1002	17
User Account Update	1002	18
User Set Server	1002	19
User Delete Address (Current)	1002	20
User Delete Server (Current)	1002	21
User Set Valid Vulnerabilities (Current)	1002	22
User Set Invalid Vulnerabilities (Current)	1002	23

Discovery and Connection Events by Type and Subtype (Continued)

EVENT NAME	EVENT TYPE	EVENT SUBTYPE
User Host Criticality	1002	24
Host Attribute Set Value (Current)	1002	25
Host Attribute Delete Value (Current)	1002	26
User Add Host	1002	27
User Add Server	1002	28
User Add Client Application	1002	29
User Add Protocol	1002	30
Reload App	1002	31
Account Delete	1002	32
Connection Statistics	1003	1
Connection Chunks	1003	2
New User Identity	1004	1
User Login	1004	2
Delete User Identity	1004	3
User Identity Dropped: User Limit Reached	1004	4
Full Host Profile	1050	N/A

TIP! For information about the data structure used for each event type/subtype, see Host Discovery Structures by Event Type on page 205.

Host Discovery Structures by Event Type

eStreamer builds host discovery event messages based on the event type indicated in the discovery event header. The following sub-sections describe the high-level structure for each event type:

- New Host and Host Last Seen Messages on page 206
- Server Messages on page 206
- New Network Protocol Message on page 207
- New Transport Protocol Message on page 208
- Client Application Messages on page 208
- IP Address Change Message on page 209
- Operating System Update Messages on page 210
- IP Address Reused and Host Timeout/Deleted Messages on page 210
- Vulnerability Change Message on page 211
- Hops Change Message on page 211
- TCP and UDP Port Closed/Timeout Messages on page 212
- MAC Address Messages on page 212
- Host Identified as a Bridge/Router Message on page 213
- VLAN Tag Information Update Messages on page 213
- Change NetBIOS Name Message on page 213
- Update Banner Message on page 214
- Policy Control Message on page 214
- User Set Vulnerabilities Messages for Version 4.6.1+ on page 216
- User Add and Delete Host Messages on page 217
- User Delete Server Message on page 217
- User Set Host Criticality Messages on page 218
- Attribute Messages on page 218
- Attribute Value Messages on page 219
- User Server and Operating System Messages on page 219
- User Protocol Messages on page 220
- User Client Application Messages on page 220
- Add Scan Result Messages on page 221
- New Operating System Messages on page 221
- Identity Conflict and Identity Timeout System Messages on page 222

The data block diagrams in the following sections depict the different record data blocks returned in host discovery event messages.

New Host and Host Last Seen Messages

New Host and Host Last Seen event messages have a standard discovery event header and a Host Profile data block (as documented in Host Profile Data Block for 5.2+ on page 343).

Note that the Host Last Seen message includes server information only for servers on the host that have changed within the Update Interval set in the discovery detection policy. In other words, only servers that have changed since the system last reported information will be included in the Host Last Seen message.

IMPORTANT! The Host Profile data block differs depending on which system version created the message. For information on legacy versions of the Host Profile data block, see Legacy Host Data Structures on page 656.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Host Profile Data Block																															

Server Messages

The following TCP and UDP server event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Server data block (as documented in Host Server Data Block 4.10.0+ on page 312):

- New TCP Server
- New UDP Server
- TCP Server Information Update
- UDP Server Information Update
- TCP Server Confidence Update
- UDP Server Confidence Update

IMPORTANT! The Server data block differs depending on which system version created the message. For information on the legacy versions of the Server data block, see Understanding Legacy Data Structures on page 457.

Each of these events use the following format:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">Server Data Block</td></tr>
</table>

New Network Protocol Message

A New Network Protocol event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a two-byte field for the network protocol (using protocol values described in the Protocol Data Block Fields table on page 244).

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="16">Network Protocol</td><td colspan="16"></td></tr>
</table>

New Transport Protocol Message

A New Transport Protocol event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) and a one-byte field for the transport protocol number (using values described in the Protocol Data Block Fields table on page 244).

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Discovery Event Header			
	Transport Protocol			

Client Application Messages

New Client Application, Client Application Update, and Client Application Timeout events have the same format and contain a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Client Application data block (see Host Client Application Data Block for 5.0+ on page 334). The discovery event header has a different record type, event type, and event subtype, depending on the event transmitted.

IMPORTANT! The Client Application data block differs depending on the system version that created the message. For information on the legacy version of the Client Application data block, see Understanding Legacy Data Structures on page 457.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Discovery Event Header			
	Client Application Data Block			

IP Address Change Message

The following host discovery messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) and two different forms, structures, one with four bytes for the IP address and one with 16 bytes for the IP address.

Four bytes are used for the IP address (in IP address octets) in the following case:

- New IPv4 to IPv4 Traffic
- Host IP Address Changed, when the RNA event version is less than 10.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">IP Address</td></tr>
</table>

16 bytes are used for the IP address in the following cases:

- New IPv6 to IPv6 Traffic
- Host IP Address Changed, when the RNA event version is 10

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">IP Address
IP Address, continued
IP Address, continued
IP Address, continued</td></tr>
</table>

Operating System Update Messages

The OS Information Update event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by an Operating System data block (as documented in Operating System Data Block 3.5+ on page 259).

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Operating System Data Block																															

IP Address Reused and Host Timeout/Deleted Messages

The following host event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) with no other data:

- Host IP Address Reused
- Host Timeout
- Host Deleted: Host Limit Reached
- Host Dropped: Host Limit Reached

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															

Vulnerability Change Message

A Vulnerability Change event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Vulnerability Reference data block (as documented in Vulnerability Reference Data Block on page 245).

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Vulnerability Reference Data Block																															

Hops Change Message

A Hops Change event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a one-byte field for the hops count.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Hops																															

TCP and UDP Port Closed/Timeout Messages

TCP and UDP Port Closed and Port Timeout event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a two-byte field for the port number.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="16">Port</td><td colspan="16"></td></tr>
</table>

MAC Address Messages

MAC Information Change and Additional MAC Detected for Host messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198), 1 byte for the TTL value, 6 bytes for the MAC address, and 1 byte to indicate whether the MAC address was detected via ARP/DHCP traffic as the actual MAC address.

IMPORTANT! If you receive MAC address messages from a system running version 4.9.x, you must check for the length of the MAC address data block and decode accordingly. If the data block is 8 bytes in length (16 bytes with the header), see MAC Address Messages on page 212. If the data block is 12 bytes in length (20 bytes with the header), see Host MAC Address 4.9+ on page 297.

Note that the MAC address data block header is **not** used within MAC Information Change and Additional MAC Detected for Host messages.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="8">TTL</td><td colspan="24" rowspan="2">MAC Address</td></tr>
<tr><td></td><td colspan="8"></td></tr>
<tr><td></td><td colspan="24">MAC Address</td><td colspan="8">ARP/DHCP</td></tr>
</table>

Host Identified as a Bridge/Router Message

A Host Identified as a Bridge/Router event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a four-byte field for the value that matches the host type:

- 0 — host
- 1 — router
- 2 — bridge

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">Host Type</td></tr>
</table>

VLAN Tag Information Update Messages

The VLAN Tag Information Update event has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by VLAN data block (as documented in VLAN Data Block on page 247).

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">VLAN Data Block</td></tr>
</table>

Change NetBIOS Name Message

A Change NetBIOS Name event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a String Information data block (as documented in String Information Data Block on page 249).

IMPORTANT! The Change NetBIOS Domain event is not currently generated by the Sourcefire 3D System.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	String Information Data Block																															

Update Banner Message

An Update Banner event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Server Banner data block (as documented in Server Banner Data Block on page 248).

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Server Banner Data Block																															

Policy Control Message

The Policy Control Message event has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Policy Control Message data block. The format of the Policy Control Message data block differs depending on the system version. For information on policy control message data block format for the current version, see Policy Engine Control Message Data Block on page 260.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">Policy Control Message Data Block</td></tr>
</table>

Connection Statistics Data Message

The Connection Statistics event has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Connection Statistics data block. The documentation of each version of the Connection Statistics data block includes the system versions that use it. For information on the connection statistics data block format for version 5.3+, see Connection Statistics Data Block 5.3+ on page 300.

IMPORTANT! The Connection Statistics data block differs depending on which system version created the message. For information on legacy versions, see the Connection Statistics data block in Understanding Legacy Data Structures on page 457.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">Connection Statistics Data Block</td></tr>
</table>

Connection Chunk Message

The Connection Chunk event has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Connection Chunk data block. The format differs depending on the system version. For information on connection chunk data block format for the current version, see Connection Chunk Data Block for 5.1.1+ on page 277.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">Connection Chunk Data Block</td></tr>
</table>

User Set Vulnerabilities Messages for Version 4.6.1+

User Set Valid Vulnerabilities, User Set Invalid Vulnerabilities, and User Vulnerability Qualification messages use the same data format: the standard discovery event header (see Discovery Event Header 5.2+ on page 198) followed by a User Vulnerability change data block (see User Vulnerability Change Data Block 4.7+ on page 285). They are differentiated by record type, event type, and event subtype.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">User Vulnerability Change Data Block</td></tr>
</table>

User Add and Delete Host Messages

The following host input event messages have the standard discovery event header (see Discovery Event Header 5.2+ on page 198) followed by a User Hosts data block (see User Hosts Data Block 4.7+ on page 283):

- User Delete Address
- User Add Hosts

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">User Hosts Data Block</td></tr>
</table>

User Delete Server Message

User Delete Server messages have the standard discovery event header (see Discovery Event Header 5.2+ on page 198) followed by a User Server List data block (see User Server List Data Block on page 281).

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">User Server List Data Block</td></tr>
</table>

User Set Host Criticality Messages

User Set Host Criticality messages have the standard discovery event header (see Discovery Event Header 5.2+ on page 198) followed by a User Criticality Change data block (see User Criticality Change Data Block 4.7+ on page 287).

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	User Criticality Change Data Block																															

Attribute Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by an Attribute Definition data block (as documented in Attribute Definition Data Block for 4.7+ on page 261):

- Add Host Attribute
- Update Host Attribute
- Delete Host Attribute

Each of these events use the following format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Attribute Definition Data Block																															

Attribute Value Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a User Attribute Value data block (as documented in User Attribute Value Data Block 4.7+ on page 289):

- Set Host Attribute Value
- Delete Host Attribute Value

Each of these events use the following format:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">User Attribute Value Data Block</td></tr>
</table>

User Server and Operating System Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a User Product data block (as documented in User Product Data Block 5.1+ on page 353):

- Set Operating System Definition
- Set Server Definition
- Add Server

Each of these events use the following format:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">User Product Data Block</td></tr>
</table>

User Protocol Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a User Protocol List data block (as documented in User Protocol List Data Block 4.7+ on page 291):

- Delete Protocol
- Add Protocol

Each of these events use the following format:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Discovery Event Header</td></tr>
<tr><td></td><td colspan="32">User Protocol List Data Block</td></tr>
</table>

User Client Application Messages

The following event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a User Client Application List data block (as documented in User Client Application List Data Block on page 268):

- Delete Client Application
- Add Client Application

Each of these events use the following format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	User Client Application List Data Block																															

Add Scan Result Messages

The Add Scan Result event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by a Scan Results data block (as documented in Scan Result Data Block 5.2+ on page 308).

This event uses the following format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Discovery Event Header																															
	Scan Result Data Block																															

New Operating System Messages

The New OS event message has a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by an Operating System Fingerprint data block (as documented in Operating System Fingerprint Data Block 5.1+ on page 339).

This event uses the following format:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Discovery Event Header			
	Operating System Fingerprint Data Block			

Identity Conflict and Identity Timeout System Messages

The Identity Conflict and Identity Timeout event messages each have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) followed by an Identity data block (as documented in Identity Data Block on page 294). These messages are generated when there are conflicts or timeouts in a fingerprint source identity.

This event uses the following format:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Discovery Event Header			
	Identity Data Block			

User Data Structures by Event Type

eStreamer builds user event messages based on the event type indicated in the discovery event header. The following sub-sections describe the high-level structure for each event type:

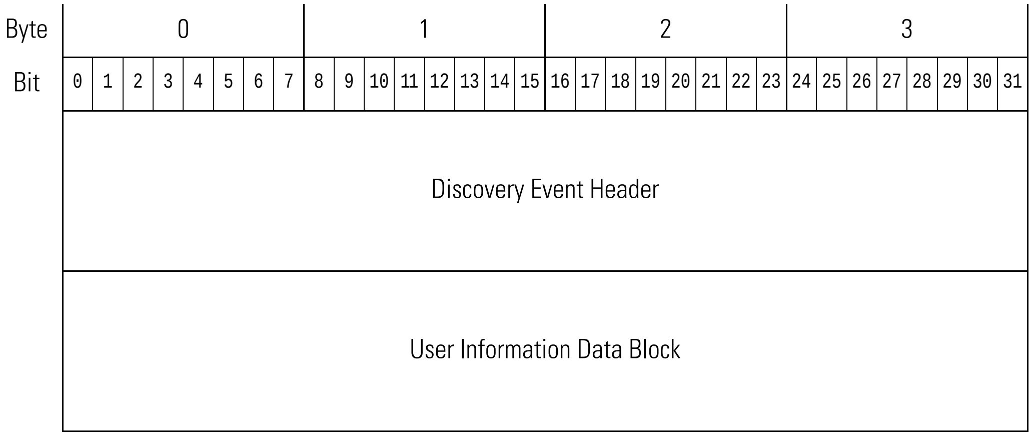
- User Modification Messages on page 223
- User Information Update Message Block on page 223

User Modification Messages

When any of the following events occurs through system detection, a user modification message is sent:

- a new user is detected (a New User Identity event—event type 1004, subtype 1),
- a user is removed (a Delete User Identity event—event type 1004, subtype 3)
- a user is dropped (a User Identity Dropped: User Limit Reached event—event type 1004, subtype 4)

User Modification event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) and a User Information data block (as documented in User Information Data Block on page 375).



User Information Update Message Block

When the login changes for a user (a User Login event—event type 1004, subtype 2) detected by the system, a user information update message is sent.

User Information Update event messages have a standard discovery event header (as documented in Discovery Event Header 5.2+ on page 198) and a User Login Information data block (as documented in User Login Information Data Block 5.1+ on page 378).

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Discovery Event Header			
	User Login Information Data Block			

Understanding Discovery (Series 1) Blocks

Most discovery and connection events incorporate one or more data blocks from the series 1 group of data structures. Each series 1 data block type conveys a particular type of information. The block type number appears in the data block header which precedes the data in the block. For information on block header format, see Data Block Header on page 46.

Series 1 Data Block Header

The series 1 data block header, like the series 2 block header, has two 32-bit integer fields that contain the block's type number and the block length.

Byte	0	1	2	3
	Data Block Type			
	Data Block Length			

IMPORTANT! The data block length field contains the number of bytes in the entire data block, including the eight bytes of the two data block header fields.

For some block series 1 types, the block header is followed immediately by raw data. In more complex block types, the header may be followed by standard fixed length fields or by the header of a series 1 primitive block that encapsulates another series 1 data block or list of blocks.

Series 1 Primitive Data Blocks

Both series 1 and series 2 blocks include a set of primitives that encapsulate lists of variable-length blocks as well as variable-length strings and BLOBs within messages. These primitive blocks have the standard series 1 block header discussed above. These primitives appear only within other series 1 data blocks. Any number can be included in a given block type. For details on the structure of the primitive blocks, see the following:

- String Data Block on page 237
- BLOB Data Block on page 238
- List Data Block on page 239
- Generic List Block on page 240

Host Discovery and Connection Data Blocks

For the list of block types in host discovery and connection events, see the Host Discovery and Connection Data Block Types table on page 225. The block types in user events are described in the User Data Block Type table on page 363. These are all series 1 data blocks.

Each entry in the table below contains a link to the subsection where the data block is defined. For each block type, the status (current or legacy) is indicated. A current data block is the latest version. A legacy data block is one that is used for an older version of the product, and the message format can still be requested from eStreamer.

Host Discovery and Connection Data Block Types

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
0	String	Current	Contains string data. See String Data Block on page 237 for more information.
1	Sub-Server	Current	Contains information about a sub-server detected on a server. See Sub-Server Data Block on page 241 for more information.
4	Protocol	Current	Contains protocol data. See Protocol Data Block on page 243 for more information.
7	Integer Data	Current	Contains integer (numeric) data. See Integer (INT32) Data Block on page 244 for more information.

Host Discovery and Connection Data Block Types (Continued)

Type	Content	Data Block Status	Description
8	Vulnerability	Current	Contains vulnerability data. See Vulnerability Reference Data Block on page 245 for more information.
10	BLOB	Current	Contains a raw block of binary data and is used specifically for banners. See BLOB Data Block on page 238 for more information.
11	List	Current	Contains a list of other data blocks. See List Data Block on page 239 for more information.
14	VLAN	Current	Contains VLAN information. See VLAN Data Block on page 247 for more information.
20	Intrusion Impact Alert	Current	Contains intrusion impact alert information. Intrusion impact alert events have slightly different headers than other data blocks. See Intrusion Impact Alert Data on page 77 for more information.
31	Generic List	Current	Contains generic list information, for example, to encapsulate lists of blocks, such as Client Application blocks, in the Host Profile block. See Generic List Block on page 240 for more information.
35	String Information	Current	Contains string information. For example, when used in the Scan Vulnerability data block, the String Information data block contains the CVE identification number data. See String Information Data Block on page 249.
37	Server Banner	Current	Contains server banner data. See Server Banner Data Block on page 248 for more information.
38	Attribute Address	Legacy	Contains the host attribute address. See Legacy Metadata Structures on page 719 for more information. The successor block is 146.

Host Discovery and Connection Data Block Types (Continued)

Type	Content	Data Block Status	Description
39	Attribute List Item	Current	Contains a host attribute list item value. See Attribute List Item Data Block on page 252 for more information.
42	Host Client Application	Legacy	Contains client application information for New Client Application events. See Host Client Application Data Block for 3.5 - 4.9.0.x on page 538 for more information.
47	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 4.8 on page 656 for more information.
48	Attribute Value	Current	Contains attribute identification numbers and values for host attributes. See Attribute Value Data Block on page 253 for more information.
51	Full Sub-Server	Current	Contains information about a sub-server detected on a server. Referenced in Full Server information blocks and in full host profiles. Includes vulnerability information for each sub-server. See Full Sub-Server Data Block on page 255 for more information.
53	Operating System	Current	Contains operating system information for Version 3.5+. See Operating System Data Block 3.5+ on page 259 for more information.
54	Policy Engine Control Message	Current	Contains information on user policy control changes. See Policy Engine Control Message Data Block on page 260 for more information.
55	Attribute Definition	Current	Contains information on attribute definitions. See Attribute Definition Data Block for 4.7+ on page 261 for more information.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
56	Connection Statistics	Legacy	Contains information for connection statistics events in 4.7 - 4.9.0. See Connection Statistics Data Block for 4.7 - 4.9.0.x on page 577 for more information.
57	User Protocol	Current	Contains protocol information from user input. See User Protocol Data Block on page 265 for more information.
59	User Client Application	Legacy	Contains client application data from user input. See User Client Application Data Block for 5.1 and earlier on page 541 for more information. Superseded by block 138.
60	User Client Application List	Current	Contains lists of user client application data blocks. See User Client Application List Data Block on page 268 for more information.
61	IP Range Specification	Legacy	Contains IP address range specifications. See IP Range Specification Data Block for 4.7.x - 5.1.1.x on page 718 for more information. Superseded by block 141.
62	Attribute Specification	Current	Contains an attribute name and value. See Attribute Specification Data Block on page 271 for more information.
63	MAC Address Specification	Current	Contains MAC address range specifications. See MAC Address Specification Data Block on page 274 for more information.
64	IP Address Specification	Current	Contains lists of IP and MAC address specification blocks. See Address Specification Data Block on page 275 for more information.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
65	User Product	Legacy	Contains host input data imported from a third party application, including third party application string mappings. See User Product Data Block for 4.10.x, 5.0 - 5.0.x on page 554 for more information. The successor block type 118 introduced for 5.0 has an identical structure as block type 65.
66	Connection Chunk	Legacy	Contains connection chunk information. See Connection Chunk Data Block for 4.10.1 - 5.1 on page 610 for more information. The successor block type 119 introduced for 5.0 has an identical structure as block type 66.
67	Fix List	Current	Contains a fix that applies to a host. See Fix List Data Block on page 279 for more information.
71	Generic Scan Results	Legacy	Contains results from an Nmap scan. See Generic Scan Results Data Block for 4.9.1.x and earlier on page 543 for more information.
72	Scan Result	Legacy	Contains results from a third-party scan. See Scan Result Data Block for 4.6.1 - 4.9.1.x on page 545 for more information.
76	User Server	Current	Contains server information from a user input event. See User Server Data Block on page 280 for more information.
77	User Server List	Current	Contains lists of user server blocks. See User Server List Data Block on page 281 for more information.
78	User Hosts	Current	Contains information about host ranges from a user host input event. See User Hosts Data Block 4.7+ on page 283 for more information.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
79	User Vulnerability	Legacy	Contains information about a vulnerability for a host or hosts. See User Vulnerability Data Block 4.7 - 4.10.x on page 563 for more information. The successor block introduced for version 5.0 has block type124.
80	User Host Vulnerability Change	Current	Contains lists of deactivated or activated vulnerabilities. See User Vulnerability Change Data Block 4.7+ on page 285 for more information.
81	User Criticality	Current	Contains information on criticality changes for a host or host. See User Criticality Change Data Block 4.7+ on page 287 for more information.
82	User Attribute Value	Current	Contains attribute value changes for a host or hosts. See User Attribute Value Data Block 4.7+ on page 289 for more information.
83	User Protocol List	Current	Contains lists of protocols for a host or hosts. See User Protocol List Data Block 4.7+ on page 291 for more information.
85	Vulnerability List	Current	Contains vulnerabilities that apply to a host. See Host Vulnerability Data Block 4.9.0+ on page 293 for more information.
86	Scan Vulnerability	Legacy	Contains information on vulnerabilities detected by a scan. See Scan Vulnerability Data Block for 4.9 - 4.9.1.x on page 551.
87	Operating System Fingerprint	Legacy	Contains lists of operating system fingerprints. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for more information. The successor block introduced for version 5.1 has block type 130.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
88	Server Information	Legacy	Contains server information used in server fingerprints. See Server Information Data Block for 4.9.1 and Earlier on page 534 for more information.
89	Host Server	Legacy	Contains server information for a host. See Host Server Data Block for 4.9.1.x on page 520 for more information.
90	Full Host Server	Legacy	Contains server information for a host. See Full Server Data Block for 4.9.0.x on page 523 for more information.
91	Host Profile	Legacy	Contains profile information for a host. See Host Profile Data Block for 5.2+ on page 343 for more information. The successor block introduced for version 5.1 has block type 132.
92	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 4.9 - 4.10.x on page 662 for more information. Supersedes data block 47.
94	Identity Data	Current	Contains identity data for a host. See Identity Data Block on page 294 for more information.
95	Host MAC Address	Current	Contains MAC address information for a host. See Host MAC Address 4.9+ on page 297 for more information.
96	Secondary Host Update	Current	Contains lists of MAC address information reported by a secondary Secondary Host Update on page 298.
97	Web Application	Legacy	Contains lists of web application data. See Web Application Data Block for 4.9.1 - 4.10.x on page 519. The successor block introduced for version 5.0 has block type 123.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
98	Host Server	Legacy	Contains server information for a host. See Host Server Data Block for 4.9.1.x on page 520 for more information.
99	Full Host Server	Legacy	Contains server information for a host. See Full Server Data Block for 4.9.1.x on page 529 for more information.
100	Host Client Application	Legacy	Contains client application information for New Client Application events. See Host Client Application Data Block for 4.9.1 - 4.10.x on page 539 for more information. The successor block type 122 introduced for version 5.0 has the same structure as block type 100.
101	Connection Statistics	Legacy	Contains information for connection statistics events in 4.9.1+. See Connection Statistics Data Block 4.9.1 - 4.10.1 on page 581 for more information.
102	Scan Results	Legacy	Contains information about a vulnerability and is used within Add Scan Result events. See Scan Result Data Block 4.10.0 - 5.1.1.x on page 548.
103	Host Server	Current	Contains server information for a host. See Host Server Data Block 4.10.0+ on page 312 for more information.
104	Full Host Server	Current	Contains server information for a host. See Full Host Server Data Block 4.10.0+ on page 314 for more information.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
105	Server Information	Legacy	Contains server information used in server fingerprints. See Server Information Data Block for 4.10.x, 5.0 - 5.0.2 on page 319 for more information. The successor block type 117 introduced for 5.0 has an identical structure as block type 105.
106	Full Server Information	Current	Contains information about a server detected on a host. See Full Server Information Data Block on page 322 for more information.
108	Generic Scan Results	Current	Contains results from an Nmap scan. See Generic Scan Results Data Block for 4.10.0+ on page 325 for more information.
109	Scan Vulnerability	Current	Contains information on vulnerabilities detected by a third-party scan. See Scan Vulnerability Data Block for 4.10.0+ on page 328.
111	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 5.0 - 5.0.2 on page 673 for more information. Supersedes data block 92.
112	Full Host Client Application	Current	Contains client application information for New Client Application events and includes a list of vulnerabilities. See Full Host Client Application Data Block 5.0+ on page 331 for more information.
115	Connection Statistics	Legacy	Contains information for connection statistics events in 5.0 - 5.0.2. See Connection Statistics Data Block 5.0 - 5.0.2 on page 590 for more information. The successor block introduced for version 5.1 has block type 126.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
117	Server Information	Current	Contains server information used in server fingerprints. See Server Information Data Block for 4.10.x, 5.0 - 5.0.2 on page 319 for more information.
118	User Product	Legacy	Contains host input data imported from a third party application, including third party application string mappings. See User Product Data Block for 4.10.x, 5.0 - 5.0.x on page 554 for more information. The predecessor block type 65, superseded in 5.0, has the same structure as this block type. The successor block introduced for version 5.1 has block type 132.
119	Connection Chunk	Legacy	Contains connection chunk information for versions 4.10.1 - 5.1. See Connection Chunk Data Block for 4.10.1 - 5.1 on page 610 for more information. The successor block is 136.
122	Host Client Application	Current	Contains client application information for New Client Application events for version 5.0+. See Host Client Application Data Block for 5.0+ on page 334 for more information. It supersedes block type 100.
123	Web Application	Current	Contains web application data for version 5.0+. See Web Application Data Block for 5.0+ on page 299 for more information. It supersedes block type 97.
124	User Vulnerability	Current	Contains information about a vulnerability for a host or hosts. See User Vulnerability Data Block 5.0+ on page 336. It supersedes block type 79.

Host Discovery and Connection Data Block Types (Continued)

TYPE	CONTENT	DATA BLOCK STATUS	DESCRIPTION
125	Connection Statistics	Legacy	Contains information for connection statistics events in 4.10.2. See Connection Statistics Data Block 4.10.2.x on page 585 for more information. The successor block introduced for version 5.1 has block type 115.
126	Connection Statistics	Legacy	Contains information for connection statistics events in 5.1. See Connection Statistics Data Block 5.1 on page 595 for more information. It supersedes block type 115. This block type is superseded by block type 137.
130	Operating System Fingerprint	Current	Contains lists of operating system fingerprints. See Operating System Fingerprint Data Block 5.1+ on page 339 for more information. It supersedes block type 87.
131	Mobile Device Information	Current	Contains information about a detected mobile device’s hardware. See Mobile Device Information Data Block for 5.1+ on page 342 for more information.
132	Host Profile	Legacy	Contains profile information for a host. See Full Host Profile Data Block 5.2.x on page 696 for more information. It supersedes block type 91. Superseded by block 139.
134	User Product	Current	Contains host input data imported from a third party application, including third party application string mappings. See User Product Data Block 5.1+ on page 353 for more information. This supersedes the predecessor block type 118.
135	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 5.1.1 on page 685 for more information. Supersedes data block 111.

Host Discovery and Connection Data Block Types (Continued)

Type	Content	Data Block Status	Description
136	Connection Chunk	Current	Contains connection chunk information. See Connection Chunk Data Block for 5.1.1+ on page 277 for more information. Supersedes block 119.
137	Connection Statistics	Legacy	Contains information for connection events in 5.1.1. See The Connection Chunk Data Block Fields table describes the components of the Connection Chunk data block: on page 611 for more information. It supersedes block type 126. It is superseded by block type 144.
138	User Client Application	Current	Contains client application data from user input. See User Client Application Data Block for 5.1.1+ on page 266 for more information. It supersedes block type 59.
139	Host Profile	Current	Contains profile information for a host. See Host Profile Data Block for 5.2+ on page 343 for more information. It supersedes block type 132.
140	Full Host Profile	Legacy	Contains complete host profile information. See Full Host Profile Data Block 5.3+ on page 388 for more information. Supersedes data block 135.
141	IP Range Specification	Current	Contains IP address range specifications. See IP Address Range Data Block for 5.2+ on page 270 for more information. It supersedes block 61.
142	Scan Results	Current	Contains information about a vulnerability and is used within Add Scan Result events. See Scan Result Data Block 5.2+ on page 308. It supersedes block 102.

Host Discovery and Connection Data Block Types (Continued)

Type	Content	Data Block Status	Description
143	Host IP	Current	Contains a host's IP address and last seen information. See Host IP Address Data Block on page 273 for more information.
144	Connection Statistics	Legacy	Contains information for connection events in 5.2.x. See Connection Statistics Data Block 5.2.x on page 602 for more information. It supersedes block type 137.
146	Attribute Address	Current	Contains the host attribute address for 5.2+. See Attribute Address Data Block 5.2+ on page 251 for more information. It supersedes block type 38.
140	Full Host Profile	Current	Contains complete host profile information. See Full Host Profile Data Block 5.3+ on page 388 for more information. Supersedes data block 135.
152	Connection Statistics	Current	Contains information for connection events in 5.3+. See Connection Statistics Data Block 5.3+ on page 300 for more information. It supersedes block type 144.

String Data Block

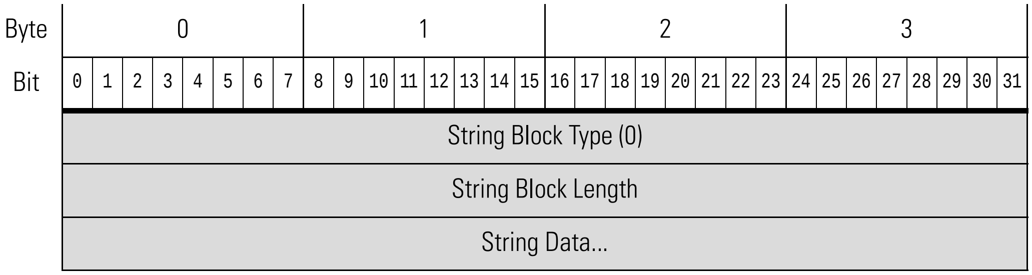
The String data block is used for sending string data in series 1 blocks. It commonly appears within other series 1 data blocks to describe, for example, operating system or server names.

Empty string data blocks (string data blocks containing no string data) have a block length value of 8 and are followed by zero bytes of string data. An empty string data block is returned when there is no content for the string value, as might happen, for example, in the OS vendor string field in an Operating System data block when the vendor of the operating system is unknown.

The String data block has a block type of 0 in the series 1 group of blocks.

IMPORTANT! Strings returned in this data block are not always null-terminated (that is, they are not always terminated with a 0).

The following diagram shows the format of the String data block:



The String Data Block Fields table describes the fields of the String data block.

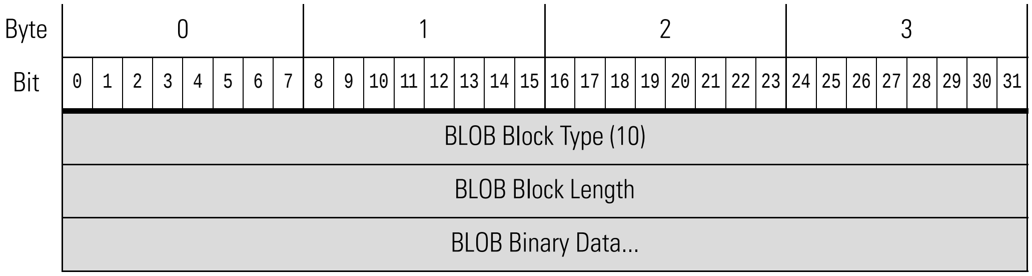
String Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block. This value is always 0.
String Block Length	uint32	Combined length of the string data block header and string data.
String Data	string	Contains the string data and may contain a terminating character (null byte) at the end of the string.

BLOB Data Block

The BLOB data block can be used to convey binary data. For example, it is used to hold the server banner captured by the system. The BLOB data block has a block type of 10 in the series 1 group of blocks.

The following diagram shows the format of the BLOB data block:



The BLOB Data Block Fields table describes the fields of the BLOB data block.

BLOB Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
BLOB Block Type	uint32	Initiates a BLOB data block. This value is always 10.
BLOB Block Length	uint32	Number of bytes in the BLOB data block, including eight bytes for the BLOB block type and length fields, plus the length of the binary data that follows.
Binary Data	variable	Contains binary data, typically a server banner.

List Data Block

The List data block is used to encapsulate a list of series 1 data blocks. For example, if a list of TCP servers is being transmitted, the Server data blocks containing the data are encapsulated in a List data block. The List data block has a block type of 11 in the series 1 group of blocks.

The following diagram shows the basic format of a List data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	List Block Type (11)																															
	List Block Length																															
	Encapsulated Data Blocks...																															

The List Data Block Fields table describes the fields of the List data block.

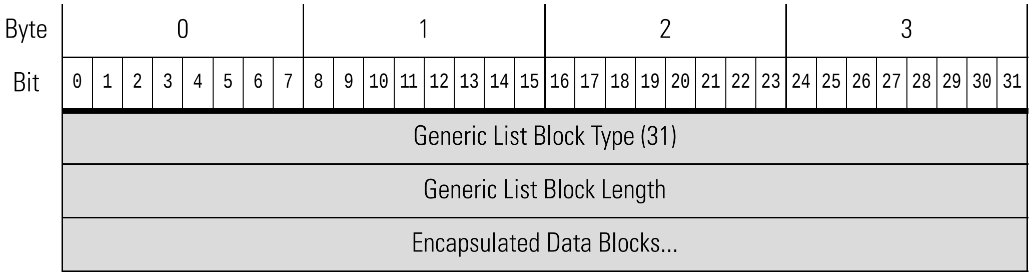
List Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
List Block Type	uint32	Initiates a List data block. This value is always 11.
List Block Length	uint32	Number of bytes in the list block and encapsulated data. For example, if there were three sub-server data blocks included in the list, the value here would include the number of bytes in the sub-server blocks, plus eight bytes for the list block header.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the list block length.

Generic List Block

The Generic List data block is used to encapsulate a list of series 1 data blocks. For example, when client application information is transmitted within a Host Profile data block, a list of Client Application data blocks are encapsulated by the Generic List data block. The Generic List data block has a block type of 31 in the series 1 group of blocks.

The following diagram shows the basic structure of a Generic List data block:



The Generic List Data Block Fields table describes the fields of the Generic List data block.

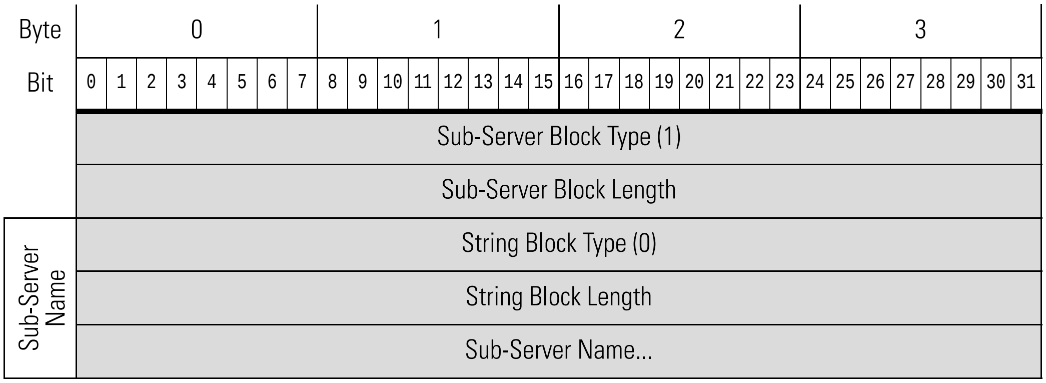
Generic List Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Encapsulated Data Blocks	variable	Encapsulated data blocks up to the maximum number of bytes in the list block length.

Sub-Server Data Block

The Sub-Server data block conveys information about an individual sub-server, which is a server called by another server on the same host and has associated vulnerabilities. The Sub-Server data block has a block type of 1 in the series 1 group of blocks.

The following diagram shows the format of the Sub-Server data block:



Vendor Name	String Block Type (0)
	String Block Length
	Vendor Name...
Version Version	String Block Type (0)
	String Block Length
	Version...

The Sub-Server Data Block Fields table describes the fields of the Sub-Server data block.

Sub-Server Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Sub-Server Block Type	uint32	Initiates a Sub-Server data block. This value is always 1.
Sub-Server Block Length	uint32	Total number of bytes in the Sub-Server data block, including eight bytes for the Sub-Server block type and length fields, plus the number of bytes of data that follows.
String Block Type	uint32	Initiates a String data block containing the sub-server name. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server name String data block, including the string block type and length fields, plus the number of bytes in the sub-server name.
Sub-Server Name	string	Name of the sub-server.
String Block Type	uint32	Initiates a String data block that contains the sub-server vendor. This value is always 0.
String Block Length	uint32	Number of bytes in the vendor name String data block, including the string block type and length fields, plus the number of bytes in the vendor name.
Vendor Name	string	Sub-server vendor name.
String Block Type	uint32	Initiates a String data block that contains the sub-server version. This value is always 0.

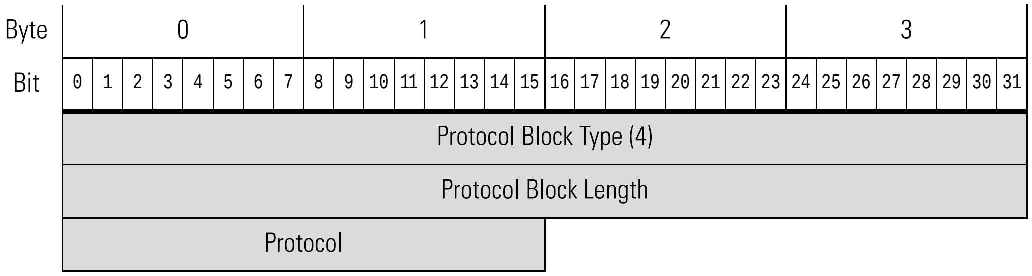
Sub-Server Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the Sub-Server version String data block, including the string block type and length fields, plus the number of bytes in the version.
Version	string	Sub-server version.

Protocol Data Block

The Protocol data block defines protocols. It is a very simple data block, with only the block type, block length, and the IANA protocol number identifying the protocol. The Protocol data block has a block type of 4 in the series 1 group of blocks.

The following graphic shows the format of the Protocol data block:



The Protocol Data Block Fields table describes the fields of the Protocol data block.

Protocol Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Protocol Block Type	uint32	Initiates a Protocol data block. This value is always 4.
Protocol Block Length	uint32	Number of bytes in the Protocol data block. This value is always 10.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP

Integer (INT32) Data Block

The Integer (INT32) data block is used in List data blocks to convey 32-bit integer data, for example, in the Vulnerability Reference data block where it is used to transmit a list of vulnerability identification numbers.

The Integer data block has a block type of 7 in the series 1 group of blocks.

The following diagram shows the format of the integer data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Integer Block Type (7)</td></tr>
<tr><td></td><td colspan="32">Integer Block Length</td></tr>
<tr><td></td><td colspan="32">Integer</td></tr>
</table>

The Integer Data Block Fields table describes the fields of the Integer data block:

Integer Data Block Fields

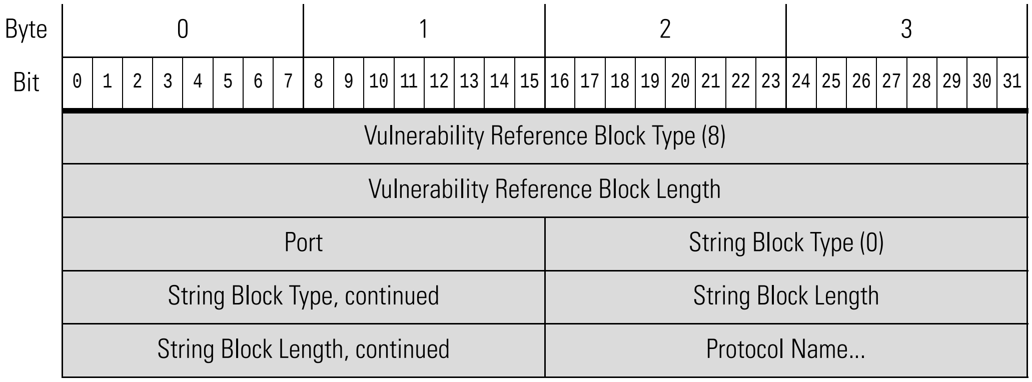
FIELD	DATA TYPE	DESCRIPTION
Integer Block Type	uint32	Initiates an Integer data block. The value is always 7.
Integer Block Length	uint32	Number of bytes in the Integer data block. This value is always 12.
Integer	uint32	Contains the integer value.

Vulnerability Reference Data Block

The Vulnerability Reference data block describes the list of vulnerabilities to which a host is subject, including the affected port, protocol, server, and list of related vulnerabilities. The Vulnerability Reference data block has a block type of 8 in the series 1 group of blocks.

IMPORTANT! An asterisk (*) next to a series 1 data block name in the following diagram indicates the message may contain zero or more instances of the block.

The following diagram shows the format of the Vulnerability Reference data block:



<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Vulnerability Reference Block Type (8)</td></tr>
<tr><td></td><td colspan="32">Vulnerability Reference Block Length</td></tr>
<tr><td></td><td colspan="16">Port</td><td colspan="16">String Block Type (0)</td></tr>
<tr><td></td><td colspan="16">String Block Type, continued</td><td colspan="16">String Block Length</td></tr>
<tr><td></td><td colspan="16">String Block Length, continued</td><td colspan="16">Protocol Name...</td></tr>
</table>

String Block Type (0)
String Block Length
Sub-Server Name...
List Block Type (11)
List Block Length
(Vulnerability ID) Integer Data Block(s) *

The Vulnerability Reference Data Block Fields table describes the fields of the Vulnerability Reference data block:

Vulnerability Reference Data Block Fields

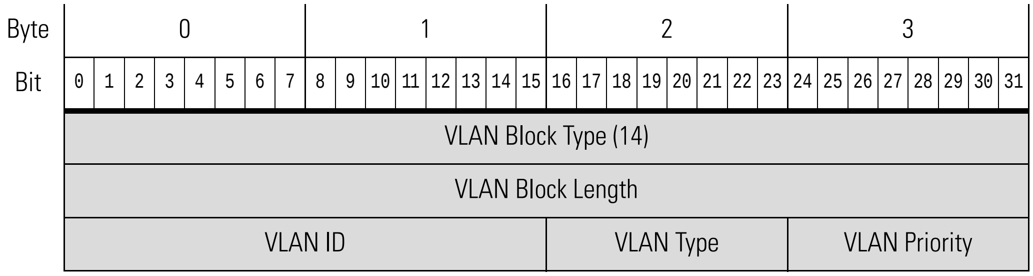
FIELD	DATA TYPE	DESCRIPTION
Vulnerability Reference Block Type	uint32	Initiates a Vulnerability Reference data block. This value is always 8.
Vulnerability Reference Block Length	uint32	Number of bytes in the Vulnerability Reference data block, including eight bytes for the vulnerability reference block type and length fields, plus the number of bytes of vulnerability reference data that follows.
Port	uint16	Port used by the sub-server affected by the listed vulnerabilities.
String Block Type	uint32	Initiates a String data block for the protocol affected by the listed vulnerabilities. This value is set to 0.
String Block Length	uint32	Number of bytes in the String data block for the protocol name, including eight bytes for the string block type and length fields, plus the number of bytes in the protocol name.
Protocol Name	string	Contains the name of the protocol used by the sub-server affected by the listed vulnerabilities.
String Block Type	uint32	Initiates a String data block for the sub-server affected by the vulnerability.

Vulnerability Reference Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the String data block containing the sub-server name, including eight bytes for the String block type and length fields, plus the number of bytes in the sub-server name.
Sub-Server	string	Contains the name of the sub-server affected by the listed vulnerabilities.
List Block Type	uint32	Initiates a list that contains zero or more VDB vulnerability ID numbers encapsulated in Integer data blocks.
List Block Length	uint32	Number of bytes in the vulnerability ID list, including eight bytes for the list block type and length fields, plus the number of bytes in the encapsulated Integer data blocks.
(Vulnerability ID) Integer Data Blocks	variable	Contains zero or more Integer data blocks containing vulnerability identification numbers. See Integer (INT32) Data Block on page 244 for the data fields that appear in an Integer data block.

VLAN Data Block

The VLAN data block contains VLAN tag information for a host. The VLAN data block has a block type of 14 in the series 1 group of blocks.The following diagram shows the format of the VLAN data block:



<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">VLAN Block Type (14)</td></tr>
<tr><td></td><td colspan="32">VLAN Block Length</td></tr>
<tr><td></td><td colspan="16">VLAN ID</td><td colspan="8">VLAN Type</td><td colspan="8">VLAN Priority</td></tr>
</table>

The VLAN Data Block Fields table describes the fields of the VLAN data block.

VLAN Data Block Fields

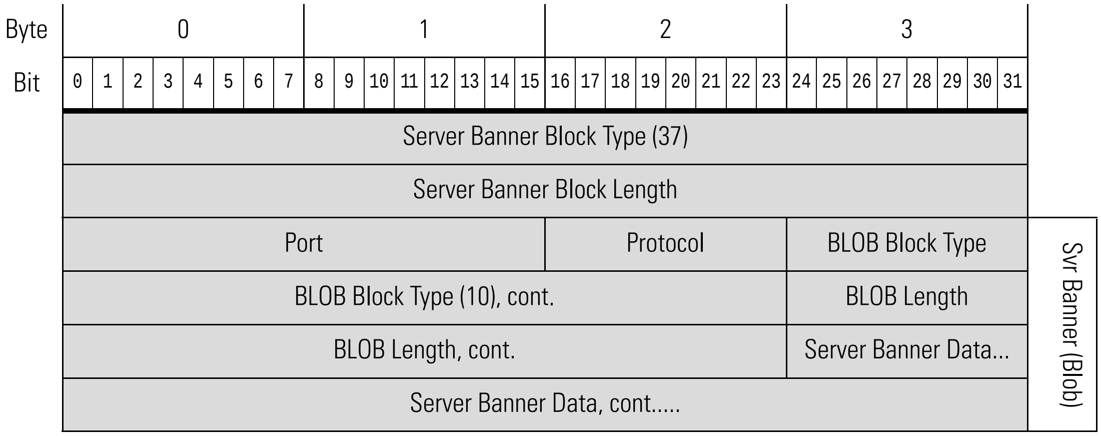
FIELD	DATA TYPE	DESCRIPTION
VLAN Block Type	uint32	Initiates a VLAN data block. This value is always 14.
VLAN Block Length	uint32	Number of bytes in the VLAN data block. This value is always 12.
VLAN ID	uint16	Contains the VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag. • 0 — Ethernet • 1 — Token Ring
VLAN Priority	uint8	Priority value included in the VLAN tag.

Server Banner Data Block

The Server Banner data block provides information about the banner for a server running on a host. It contains the server port, protocol, and the banner data. The Server Banner data block has a block type of 37 in the series 1 group of blocks.

The following diagram shows the format of the Server Banner data block.

IMPORTANT! An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.



The Server Banner Data Block Fields table describes the fields of the Server Banner data block.

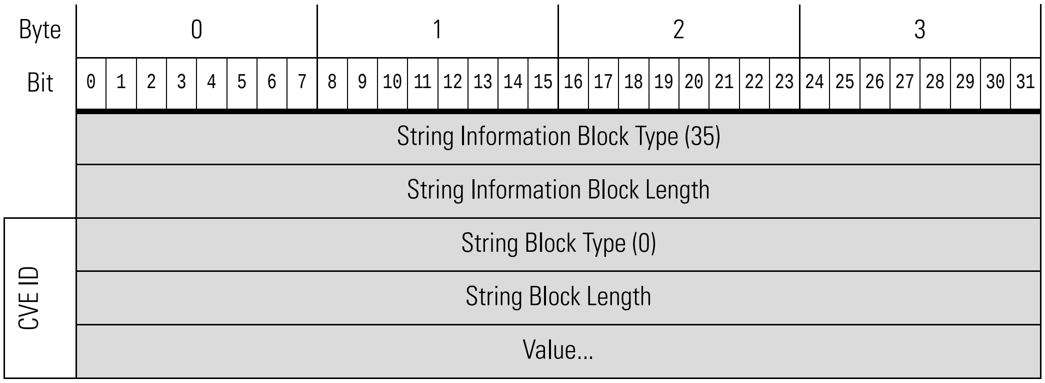
Server Banner Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Server Banner Block Type	uint32	Initiates a Server Banner data block. This value is always 37.
Server Banner Block Length	uint32	Total number of bytes in the Server Banner data block, including the eight bytes in the server banner block type and length fields, plus the number of bytes of data that follows.
Port	uint16	Port number on which the server runs.
Protocol	uint8	Protocol number for the server.
BLOB Block Type	uint32	Initiates a BLOB data block containing server banner data. This value is always 10.
Length	uint32	Total number of bytes in the BLOB data block (typically 264 bytes).
Banner	byte[*n*]	First *n* bytes of the packet involved in the server event, where *n* is equal to or less than 256.

String Information Data Block

The String Information data block contains string data. For example, the String Information data block is used to convey the Common Vulnerabilities and Exposures (CVE) identification string within a Scan Vulnerability data block. The String Information data block has a block type of 35 in the series 1 group of blocks.

The following diagram shows the format of the String Information data block:



	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		String Information Block Type (35)			
		String Information Block Length			
CVE ID		String Block Type (0)			
		String Block Length			
		Value...			

The String Information Data Block Fields table describes the fields of the String Information data block.

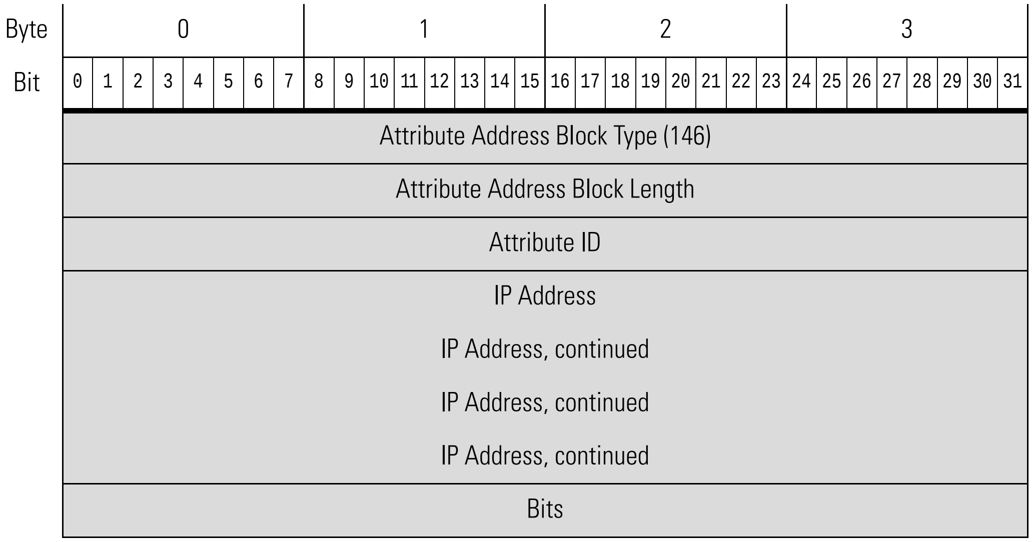
String Information Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
String Information Block Type	uint32	Initiates a String Information data block. This value is always 35.
String Information Block Length	uint32	Combined length of the String Information data block header and String Information data.
String Block Type	uint32	Initiates a string data block for the value.
String Block Length	uint32	Number of bytes in the string data block for the value, including eight bytes for the string block type and length, plus the number of bytes in the value.
Value	string	The value of the Common Vulnerabilities and Exposures (CVE) identification number for the vulnerability data block where the String Information data block is used.

Attribute Address Data Block 5.2+

The Attribute Address data block contains an attribute list item and is used within an Attribute Definition data block. It has a block type of 146 in the series 1 group of blocks.

The following diagram shows the basic structure of an Attribute Address data block:



Attribute Address Block Type (146)

Attribute Address Block Length

Attribute ID

IP Address

IP Address, continued

IP Address, continued

IP Address, continued

Bits

The Attribute Address Data Block 5.2+ Fields table describes the fields of the Attribute Address data block.

Attribute Address Data Block 5.2+ Fields

FIELD	DATA TYPE	DESCRIPTION
Attribute Address Block Type	uint32	Initiates an Attribute Address data block. This value is always 146.
Attribute Address Block Length	uint32	Number of bytes in the Attribute Address data block, including eight bytes for the attribute address block type and length, plus the number of bytes in the attribute address data that follows.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.

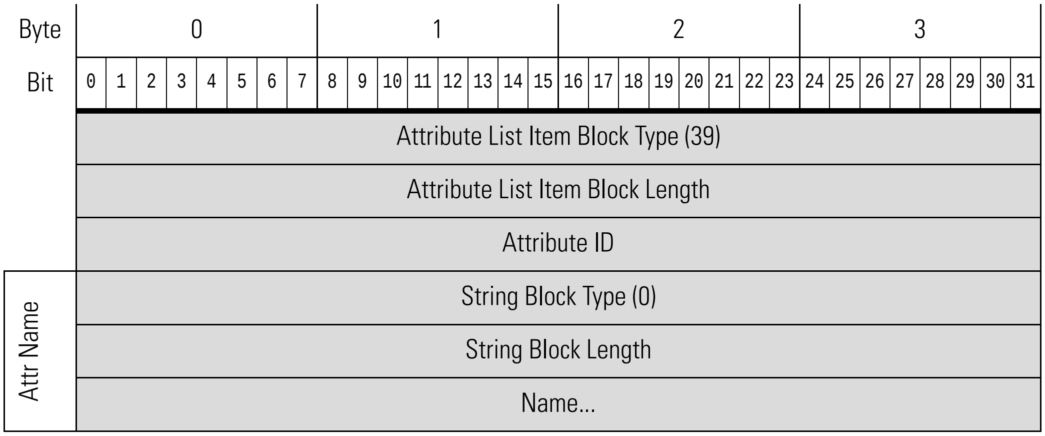
Attribute Address Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
IP Address	uint8[16]	IP address of the host, if the address was automatically assigned. The address can be IPv4 or IPv6.
Bits	uint32	Contains the significant bits used to calculate the netmask if an IP address was automatically assigned.

Attribute List Item Data Block

The Attribute List Item data block contains an attribute list item and is used within an Attribute Definition data block. It has a block type of 39 in the series 1 group of blocks.

The following diagram shows the basic structure of an Attribute List Item data block:



Byte 0 1 2 3

Bit 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Attribute List Item Block Type (39)

Attribute List Item Block Length

Attribute ID

Attr Name: String Block Type (0)

String Block Length

Name...

The Attribute List Item Data Block Fields table describes the fields of the Attribute List Item data block.

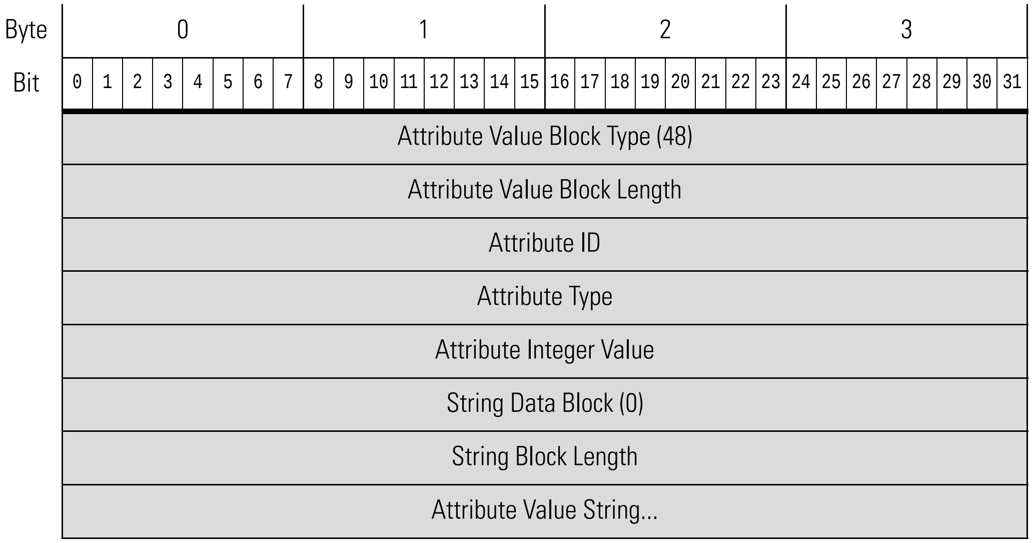
Attribute List Item Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Attribute List Item Block Type	uint32	Initiates an Attribute List Item data block. This value is always 39.
Attribute List Item Block Length	uint32	Number of bytes in the Attribute List Item data block, including eight bytes for the attribute list item block type and length, plus the number of bytes in the attribute list item data that follows.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
String Block Type	uint32	Initiates a String data block for the attribute list item name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the attribute list item name, including eight bytes for the string block type and length, plus the number of bytes in the attribute list item name.
Name	string	Attribute list item name.

Attribute Value Data Block

The Attribute Value data block conveys attribute identification numbers and values for host attributes. An Attribute Value data block for each attribute applied to the host in the event is included in a list in the Full Host Profile data block. The Attribute Value data block has a block type of 48 in the series 1 group of blocks.

The following diagram shows the format of the Attribute Value data block:



The Attribute Value Data Block Fields table describes the components of the Attribute Value data block.

Attribute Value Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Attribute Value Block Type	uint32	Initiates an Attribute Value data block. This value is always 48.
Attribute Value Block Length	uint32	Total number of bytes in the Attribute Value data block, including eight bytes for the attribute value block type and length fields, plus the number of bytes of attribute block data that follows.
Attribute ID	uint32	The identification number for the attribute.

Attribute Value Data Block Fields (Continued)

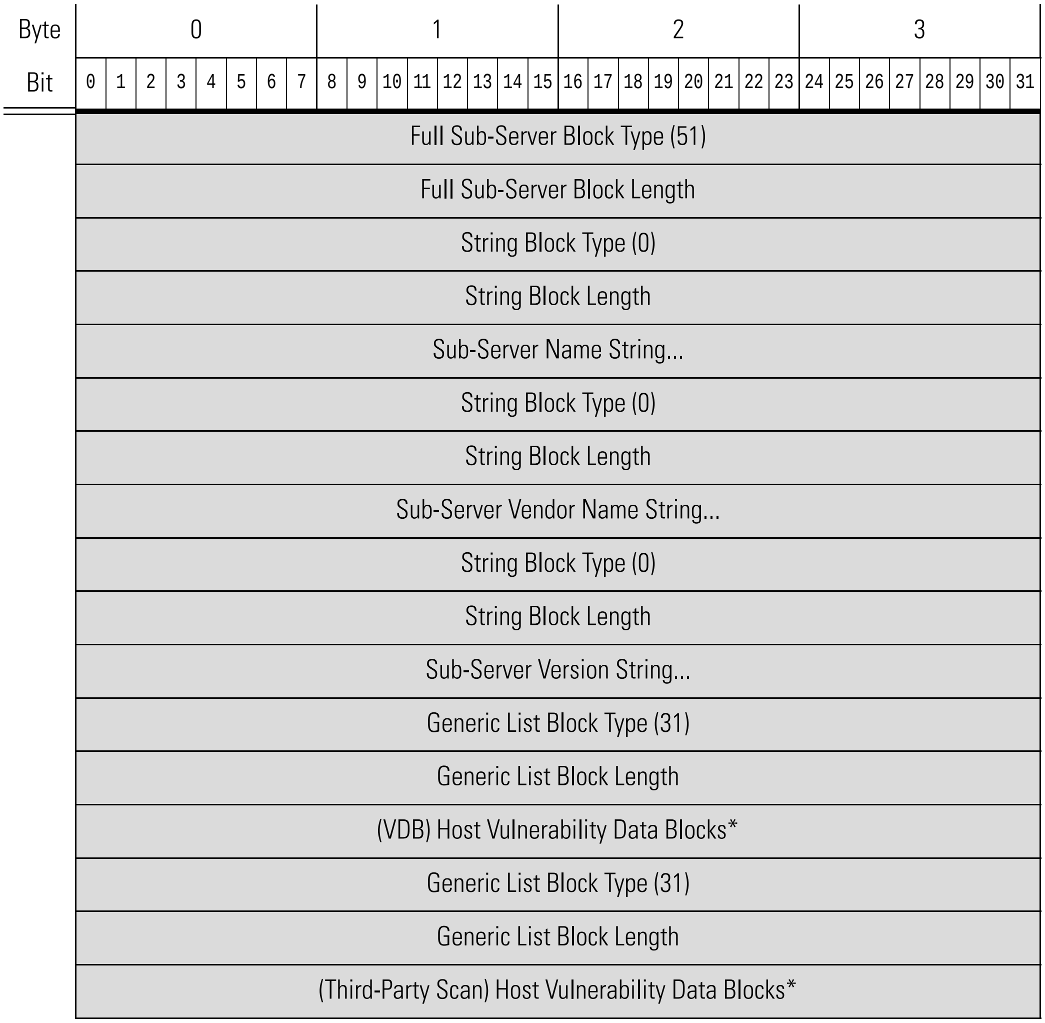
FIELD	DATA TYPE	DESCRIPTION
Attribute Type	uint32	Type of affected attribute. Possible values are: • `0` — attribute with text as value; this uses string data • `1` — attribute with value in range; this uses integer data • `2` — attribute with a list of possible values, this uses integer data • `3` — attribute with a URL as value; this uses string data • `4` — attribute with binary BLOB as value; this uses string data
Attribute Integer Value	uint32	Integer value for the attribute, if applicable.
String Block Type	uint32	Initiates a String data block containing the attribute name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including the string block type and length fields, plus the number of bytes in the attribute name.
Attribute Value	string	Value of the attribute.

Full Sub-Server Data Block

The Full Sub-Server data block conveys information about a sub-server associated with a server detected on a host, and includes information about the sub-server such as its vendor and version and any related Sourcefire-VDB and third-party vulnerabilities for the sub-server on the host. A sub-server is a loadable module of a server that has its own associated vulnerabilities. A Full Host Server data block includes a Full Sub-Server data block for each sub-server detected on the host. The Full Sub-Server data block has a block type of 51 in the series 1 group of blocks.

IMPORTANT! An asterisk (*) next to a series 1 data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Sub-Server data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Sub-Server Block Type (51)			
	Full Sub-Server Block Length			
	String Block Type (0)			
	String Block Length			
	Sub-Server Name String...			
	String Block Type (0)			
	String Block Length			
	Sub-Server Vendor Name String...			
	String Block Type (0)			
	String Block Length			
	Sub-Server Version String...			
	Generic List Block Type (31)			
	Generic List Block Length			
	(VDB) Host Vulnerability Data Blocks*			
	Generic List Block Type (31)			
	Generic List Block Length			
	(Third-Party Scan) Host Vulnerability Data Blocks*			

The Full Sub-Server Data Block Fields table describes the components of the Full Sub-Server data block.

Full Sub-Server Data Block Fields

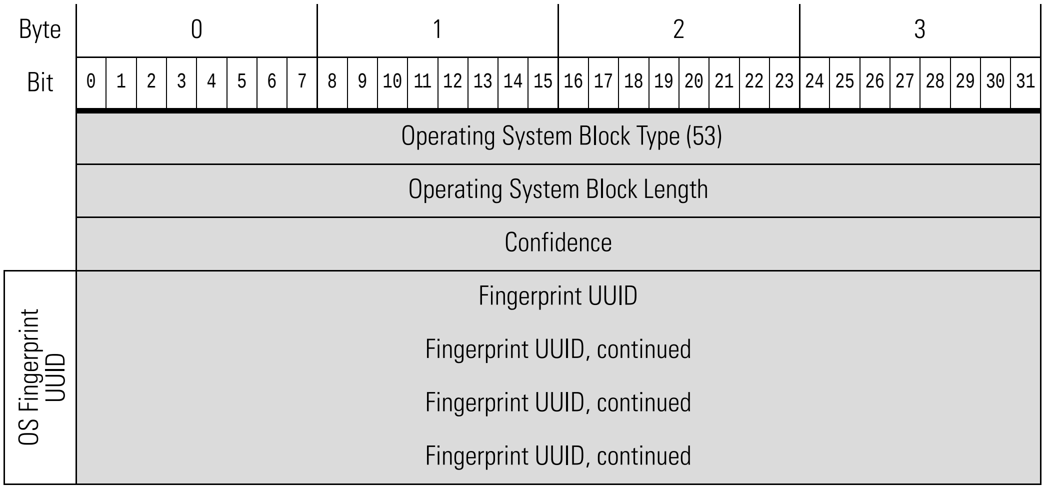
FIELD	DATA TYPE	DESCRIPTION
Full Sub-Server Block Type	uint32	Initiates a Full Sub-Server data block. This value is always 51.
Full Sub-Server Block Length	uint32	Total number of bytes in the Full Sub-Server data block, including eight bytes for the Full Sub-Server block type and length fields, plus the number of bytes in the full sub-server data that follows.
String Block Type	uint32	Initiates a String data block containing the sub-server name. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server name.
Sub-Server Name	string	Sub-server name.
String Block Type	uint32	Initiates a String data block containing the sub-server vendor's name. This value is always 0.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server vendor name.
Sub-Server Vendor Name	string	Name of the sub-server vendor.
String Block Type	uint32	Initiates a String data block that contains the sub-server version. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server version.
Sub-Server Version	string	Sub-server version.

Full Sub-Server Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB Vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Sourcefire-VDB Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities identified by Sourcefire. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Third-Party Scan Vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third-Party Scan Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities identified by a third-party vulnerability scanner. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.

Operating System Data Block 3.5+

The operating system data block for Version 3.5+ has a block type of 53 in the series 1 group of blocks. The block includes a fingerprint Universally Unique Identifier (UUID). The following diagram shows the format of an operating system data block in 3.5+.


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Operating System Block Type (53)
Operating System Block Length
Confidence
OS Fingerprint UUID
Fingerprint UUID
Fingerprint UUID, continued
Fingerprint UUID, continued
Fingerprint UUID, continued

The Operating System Data Block 3.5+ Fields table describes the fields of the v3.5 operating system data block.

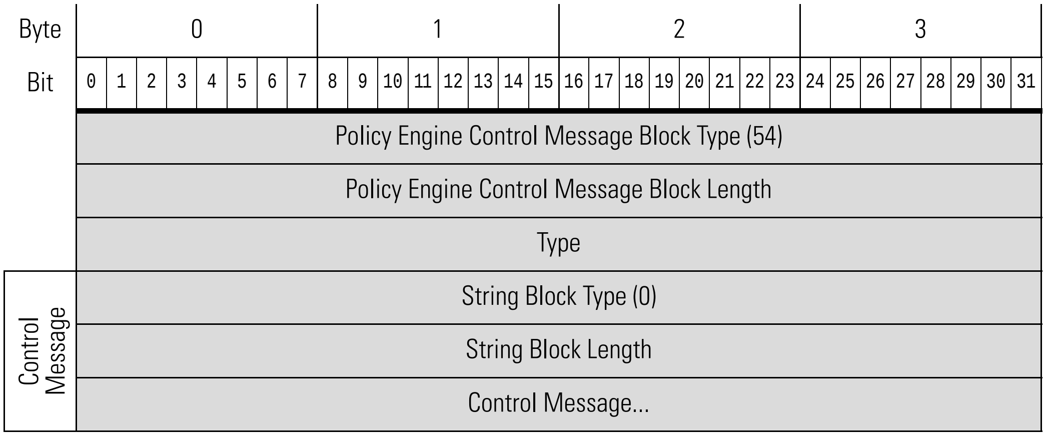
Operating System Data Block 3.5+ Fields

FIELD	DATA TYPE	DESCRIPTION
Operating System Data Block Type	uint32	Initiates the operating system data block. This value is always 53.
Operating System Data Block Length	uint32	Number of bytes in the Operating System data block. This value should always be 28: eight bytes for the data block type and length fields, plus four bytes for the confidence value and sixteen bytes for the fingerprint UUID value.
Confidence	uint32	Confidence percentage value.
Fingerprint UUID	uint8[16]	Fingerprint identification number, in octets, that acts as a unique identifier for the operating system. The fingerprint UUID maps to the operating system name, vendor, and version in the Sourcefire database.

Policy Engine Control Message Data Block

The Policy Engine Control Message data block conveys the control message content for policy types. The Policy Engine Control Message data block has a block type of 54 in the series 1 group of blocks.

The following diagram shows the format of the Policy Engine Control Message data block:



The Policy Engine Control Message Data Block Fields table describes the components of the Policy Engine Control Message data block.

Policy Engine Control Message Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Policy Engine Control Message Block Type	uint32	Initiates a Policy Engine Control Message data block. This value is always 54.
Policy Engine Control Message Length	uint32	Total number of bytes in the Policy Engine Control Message data block, including eight bytes for the policy engine control block type and length fields, plus the number of bytes of policy engine control data that follows.
Type	uint32	Indicates the type of policy for the event.
String Block Type	uint32	Initiates a String data block that contains the control message. This value is always 0.

Policy Engine Control Message Data Block Fields (Continued)

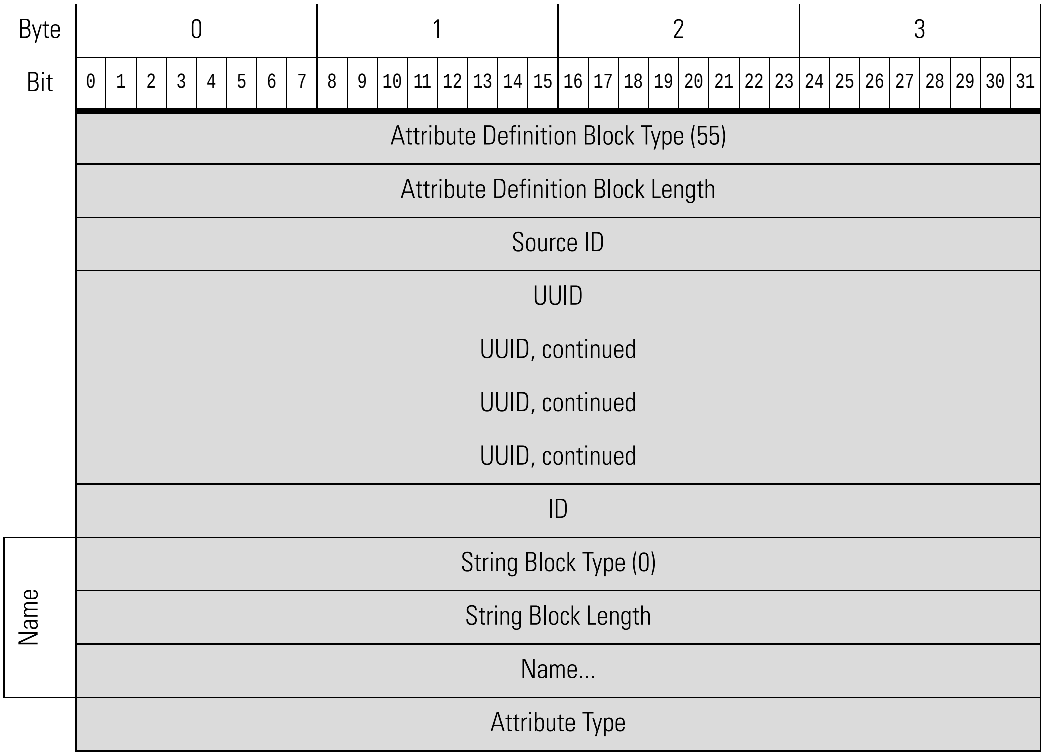
FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the control message String data block, including eight bytes for the block type and length fields, plus the number of bytes in the control message.
Control Message	uint32	The control message from the policy engine.

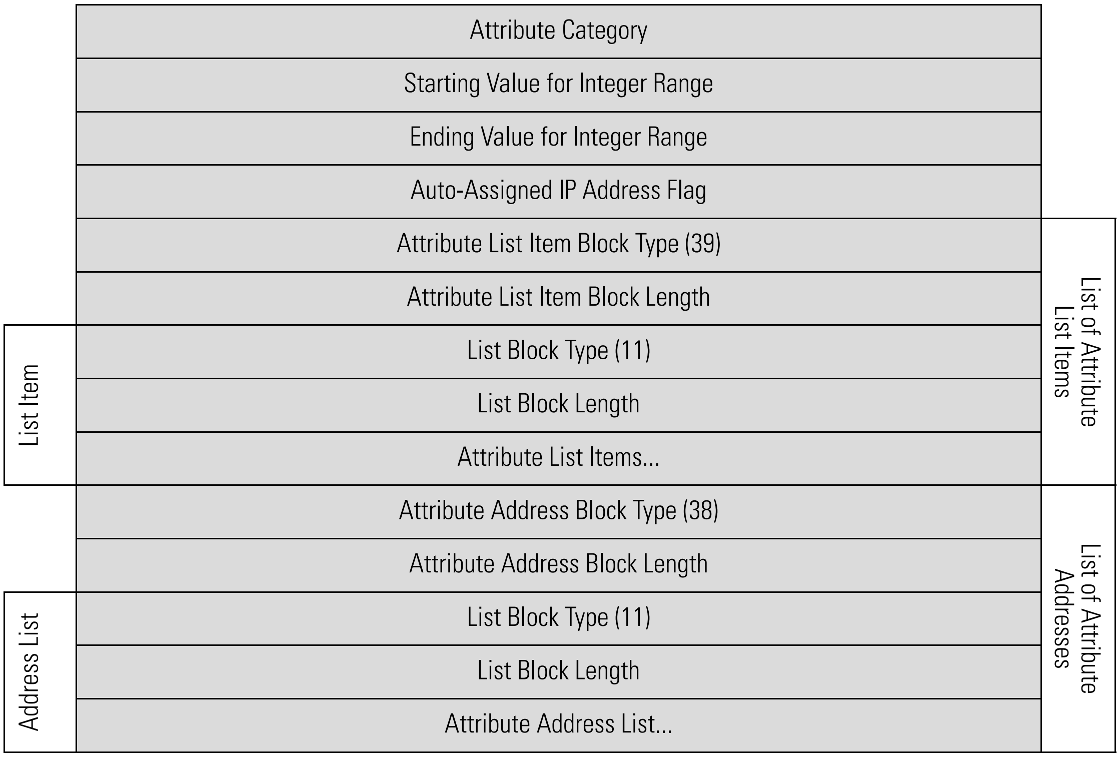
Attribute Definition Data Block for 4.7+

The Attribute Definition data block contains the attribute definition in an attribute creation, change, or deletion event and is used within Host Attribute Add events (event type 1002, subtype 6), Host Attribute Update events (event type 1002, subtype 7), and Host Attribute Delete events (event type 1002, subtype 8). It has a block type of 55 in the series 1 group of blocks.

For more information on those events, see Attribute Messages on page 218.

The following diagram shows the basic structure of an Attribute Definition data block:




Attribute Category
Starting Value for Integer Range
Ending Value for Integer Range
Auto-Assigned IP Address Flag
Attribute List Item Block Type (39)
Attribute List Item Block Length
List Block Type (11)
List Block Length
Attribute List Items...
Attribute Address Block Type (38)
Attribute Address Block Length
List Block Type (11)
List Block Length
Attribute Address List...
List Item
Address List
List of Attribute List Items
List of Attribute Addresses

The Attribute Definition Data Block Fields table describes the fields of the Attribute Definition data block.

Attribute Definition Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Attribute Definition Block Type	uint32	Initiates an Attribute Definition data block. This value is always 55.
Attribute Definition Block Length	uint32	Number of bytes in the Attribute Definition data block, including eight bytes for the attribute definition block type and length, plus the number of bytes in the attribute definition data that follows.
Source ID	uint32	Identification number that maps to the source of the attribute data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.

Attribute Definition Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
UUID	uint8[16]	An ID number that acts as a unique identifier for the affected attribute.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
String Block Type	uint32	Initiates a String data block for the attribute definition name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the attribute definition name, including eight bytes for the string block type and length, plus the number of bytes in the attribute definition name.
Name	string	Attribute definition name.
Attribute Type	uint32	Type of attribute. Possible values are: • `0` — attribute with text as value; this uses string data • `1` — attribute with value in range; this uses integer data • `2` — attribute with a list of possible values; this uses integer data • `3` — attribute with a URL as value; this uses string data • `4` — attribute with binary BLOB as value; this uses string data
Attribute Category	uint32	Attribute category.
Starting Value for Range	uint32	First integer in the integer range for the defined attribute.
Ending Value for Range	uint32	Last integer in the integer range for the defined attribute.
Auto-Assigned IP Address Flag	uint32	Flag indicating if an IP address is auto-assigned based on the attribute.
List Block Type	uint32	Initiates a List data block comprising Attribute List Item data blocks conveying attribute list items. This value is always 11.

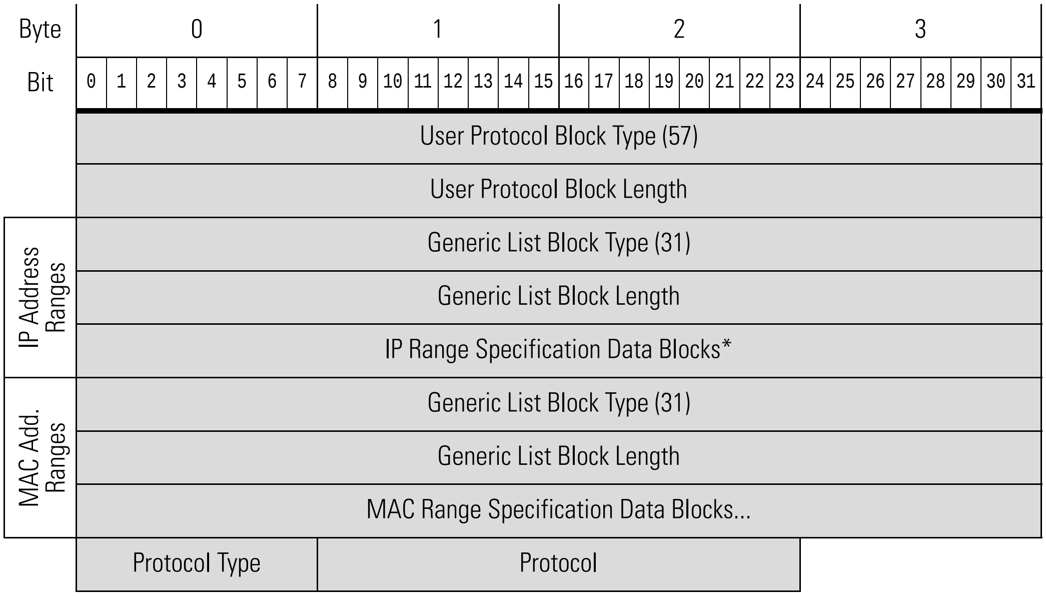
Attribute Definition Data Block Fields (Continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Attribute List Item data blocks. This field is followed by zero or more Attribute List Item data blocks.
Attribute List Item Block Type	uint32	Initiates the first Attribute List Item data block. This data block can be followed by other Attribute List Item data blocks up to the limit defined in the list block length field.
Attribute List Item Block Length	uint32	Number of bytes in the Attribute List Item String data block, including eight bytes for the block type and header fields, plus the number of bytes in the attribute list item.
Attribute List Item	variable	Attribute List Item data as documented in Attribute List Item Data Block on page 252.
List Block Type	uint32	Initiates a List data block comprising Attribute Address data blocks conveying IP addresses for hosts with the attribute. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Attribute Address data blocks. This field is followed by zero or more Attribute Address data blocks.
Attribute Address Block Type	uint32	Initiates the first Attribute Address data block. This data block can be followed by other Attribute Address data blocks up to the limit defined in the list block length field.
Attribute Address Block Length	uint32	Number of bytes in the Attribute Address data block, including eight bytes for the block type and header fields, plus the number of bytes in the attribute address.
Attribute Address	variable	Attribute Address data as documented in Attribute Address Data Block 5.2+ on page 251.

User Protocol Data Block

The User Protocol data block is used to contain information about added protocols, the type of the protocol, and lists of IP address and MAC address ranges for the hosts with the protocol. The User Protocol data block has a block type of 57 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Protocol data block:


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
User Protocol Block Type (57)
User Protocol Block Length
IP Address Ranges
Generic List Block Type (31)
Generic List Block Length
IP Range Specification Data Blocks*
MAC Add. Ranges
Generic List Block Type (31)
Generic List Block Length
MAC Range Specification Data Blocks...
Protocol Type
Protocol

The User Protocol Data Block Fields table describes the fields of the User Protocol data block.

User Protocol Data Block Fields

Field	Number of Bytes	Description
User Protocol Block Type	uint32	Initiates a User Protocol data block. This value is always 57.
User Protocol Block Length	uint32	Total number of bytes in the User Protocol data block, including eight bytes for the user protocol block type and length fields, plus the number of bytes of user protocol data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.

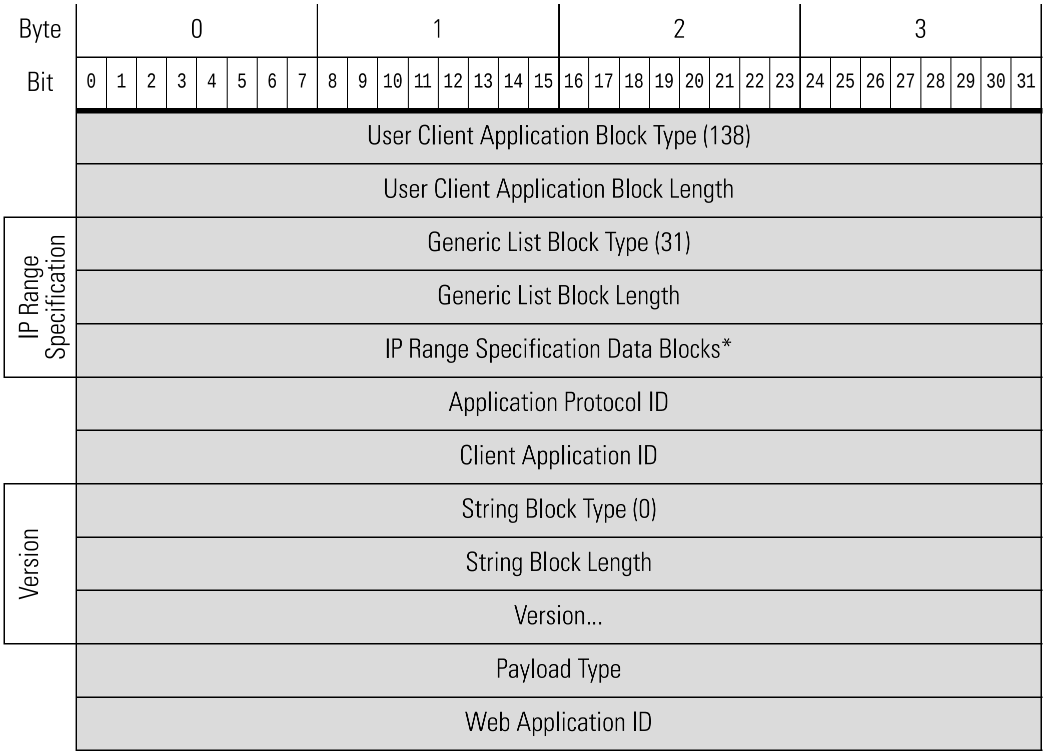
User Protocol Data Block Fields (Continued)

FIELD	NUMBER OF BYTES	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising MAC Range Specification data blocks conveying MAC address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated MAC Range Specification data blocks.
MAC Range Specification Data Blocks *	variable	MAC Range Specification data blocks containing information about the MAC address ranges for the user input. See MAC Address Specification Data Block on page 274 for a description of this data block.
Protocol Type	uint8	Indicates the type of the protocol. The protocol can be either `0`, for a network layer protocol such as IP, or `1` for a transport layer protocol such as TCP or UDP.
Protocol	uint16	Indicates the protocol for the data contained in the data block.

User Client Application Data Block for 5.1.1+

The User Client Application data block contains information about the source of the client application data, the identification number for the user who added the data, and the lists of IP address range data blocks. The payload ID, which was added in Version 5.3, specifies the application instance associated with the record. The User Client Application data block has a block type of 138 in the series 1 group of blocks. It replaces block type 59.

The following diagram shows the basic structure of a User Client Application data block:



The User Client Application Data Block Fields table describes the fields of the User Client Application data block.

User Client Application Data Block Fields

Field	Number of Bytes	Description
User Client Application Block Type	uint32	Initiates a User Client Application data block. This value is always 138.
User Client Application Block Length	uint32	Total number of bytes in the User Client Application data block, including eight bytes for the user client application block type and length fields, plus the number of bytes of user client application data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.

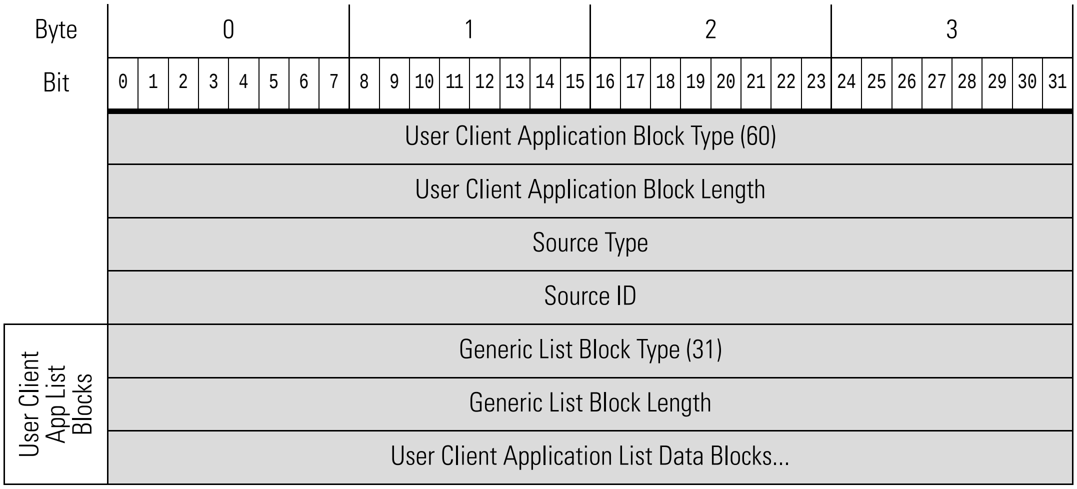
User Client Application Data Block Fields (Continued)

FIELD	NUMBER OF BYTES	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block that contains the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the client application version String data block, including the string block type and length fields, plus the number of bytes in the version.
Version	string	Client application version.
Payload Type	uint32	This field is included for backwards compatibility. It is always `0`.
Web Application ID	uint32	The internal identification number for the web application, if applicable.

User Client Application List Data Block

The User Client Application List data block contains information about the source of the client application data, the identification number for the user who added the data, and the lists of client application blocks. The User Client Application List data block has a block type of 60 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Client Application List data block:


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
User Client Application Block Type (60)
User Client Application Block Length
Source Type
Source ID
User Client App List Blocks
Generic List Block Type (31)
Generic List Block Length
User Client Application List Data Blocks...

The User Client Application List Data Block Fields table describes the fields of the User Client Application List data block.

User Client Application List Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
User Client Application List Block Type	uint32	Initiates a User Client Application List data block. This value is always 60.
User Client Application List Block Length	uint32	Total number of bytes in the User Client Application List data block, including eight bytes for the user client application list block type and length fields, plus the number of bytes of user client application list data that follows.
Source Type	uint32	Number that maps to the type of data source: • `0` if the client data was detected by RNA • `1` if the client data was provided by a user • `2` if the client data was detected by a third-party scanner • `3` if the client data was provided by a command line tool such as nmimport.pl or the Host Input API client

User Client Application List Data Block Fields (Continued)

Field	Number of Bytes	Description
Source ID	uint32	Identification number that maps to the source that added the affected client application. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Client Application Blocks	variable	Encapsulated User Client Application data blocks up to the maximum number of bytes in the list block length. For more information on the User Client Application data block, see User Client Application Data Block for 5.1.1+ on page 266.

IP Address Range Data Block for 5.2+

The IP Address Range data block for 5.2+ conveys a range of IP addresses. IP Address Range data blocks are used in User Protocol, User Client Application, Address Specification, User Product, User Server, User Hosts, User Vulnerability, User Criticality, and User Attribute Value data blocks. The IP Address Range data block has a block type of 141 in the series 1 group of blocks.

The following diagram shows the format of the IP Address Range data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	IP Address Range Block Type (141)			
	IP Address Range Block Length			
	IP Address Range Start			
	IP Address Range Start, continued			

IP Address Range Start, continued
IP Address Range Start, continued
IP Address Range End
IP Address Range End, continued
IP Address Range End, continued
IP Address Range End, continued

The IP Address Range Data Block Fields table describes the components of the IP Address Range Specification data block.

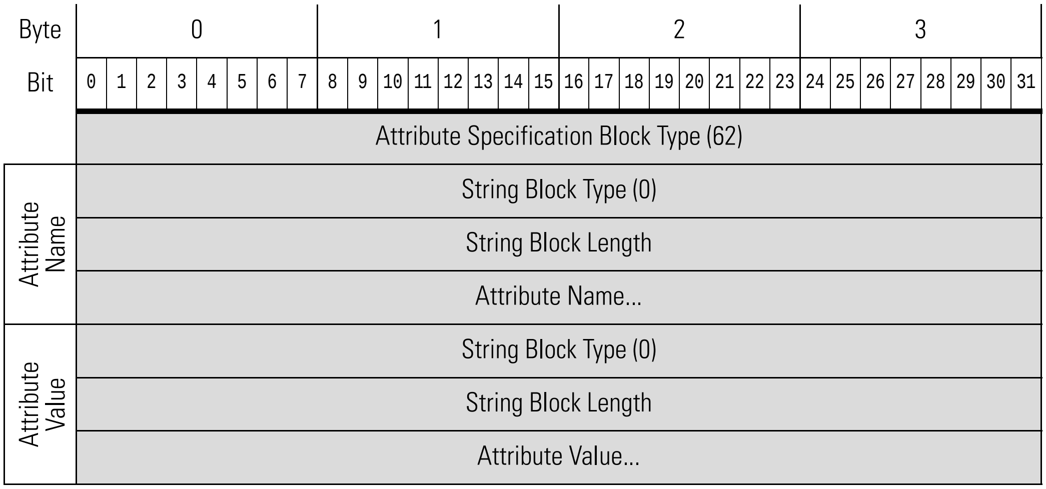
IP Address Range Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
IP Address Range Block Type	uint32	Initiates a IP Address Range data block. This value is always 61.
IP Address Range Block Length	uint32	Total number of bytes in the IP Address Range data block, including eight bytes for the IP Address Range block type and length fields, plus the number of bytes of IP Address Range data that follows.
IP Address Range Start	uint8[16]	The starting IP address for the IP address range.
IP Address Range End	uint8[16]	The ending IP address for the IP address range.

Attribute Specification Data Block

The Attribute Specification data block conveys the attribute name and value. The Attribute Specification data block has a block type of 62 in the series 1 group of blocks.

The following diagram shows the format of the Attribute Specification data block:



The Attribute Specification Data Block Fields table describes the components of the Attribute Specification data block.

Attribute Specification Data Block Fields

Field	Data Type	Description
Attribute Specification Block Type	uint32	Initiates an Attribute Specification data block. This value is always 62.
String Block Type	uint32	Initiates a String data block that contains the attribute name. This value is always 0.
String Block Length	uint32	Number of bytes in the attribute name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the attribute name.
Attribute Value	uint32	The value of the attribute.
String Block Type	uint32	Initiates a String data block that contains the attribute name. This value is always 0.

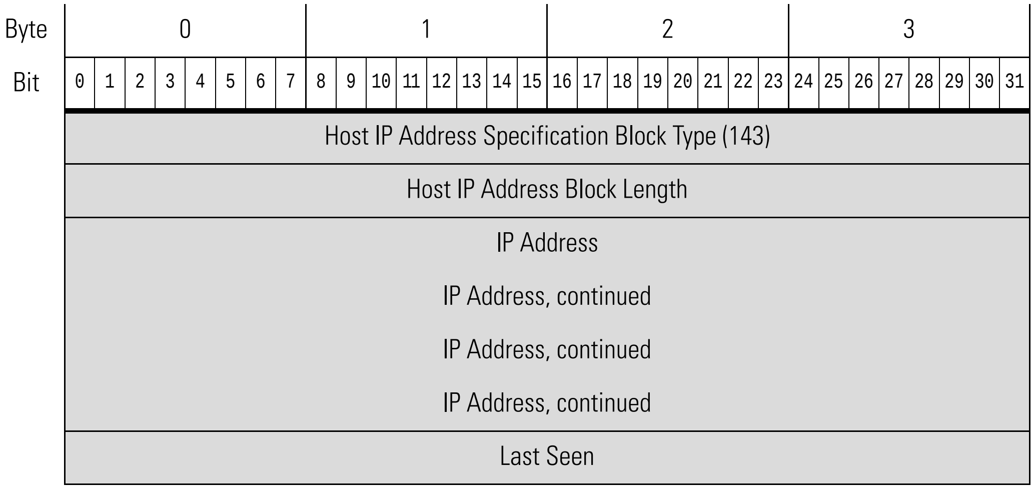
Attribute Specification Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the attribute name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the attribute name.
Attribute Name	uint32	The name of the attribute.

Host IP Address Data Block

The Host IP Address data block conveys an individual IP address. The IP address may be either an IPv4 or IPv6 address. Host IP Address data blocks are used in User Protocol, Address Specification, and User Host data blocks. The Host IP data block has a block type of 143 in the series 1 group of blocks.

The following diagram shows the format of the Host IP Address data block:



The Host IP Address Data Block Fields table describes the components of the Host IP Address data block.

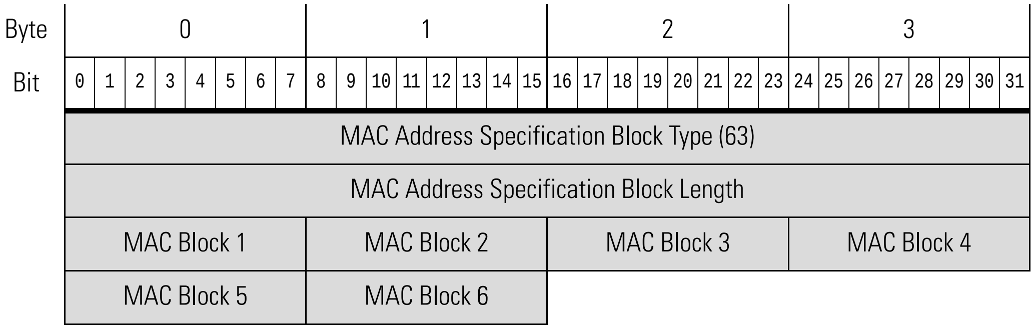
Host IP Address Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Host IP Address Block Type	uint32	Initiates a Host IP Address data block. This value is always 143.
Host IP Block Length	uint32	Total number of bytes in the Host IP Address data block, including eight bytes for the Host IP block type and length fields, plus the number of bytes of Host IP Address data that follows.
IP Address	uint8[16]	The IP address. This can be IPv4 or IPv6.
Last Seen	uint32	UNIX timestamp that represents the last time the IP address was detected.

MAC Address Specification Data Block

The MAC Address Specification data block conveys an individual MAC address. MAC Address Specification data blocks are used in User Protocol, Address Specification, and User Hosts data blocks. The MAC Address Specification data block has a block type of 63 in the series 1 group of blocks.

The following diagram shows the format of the MAC Address Specification data block:



The MAC Address Specification Data Block Fields table describes the components of the MAC Address Specification data block.

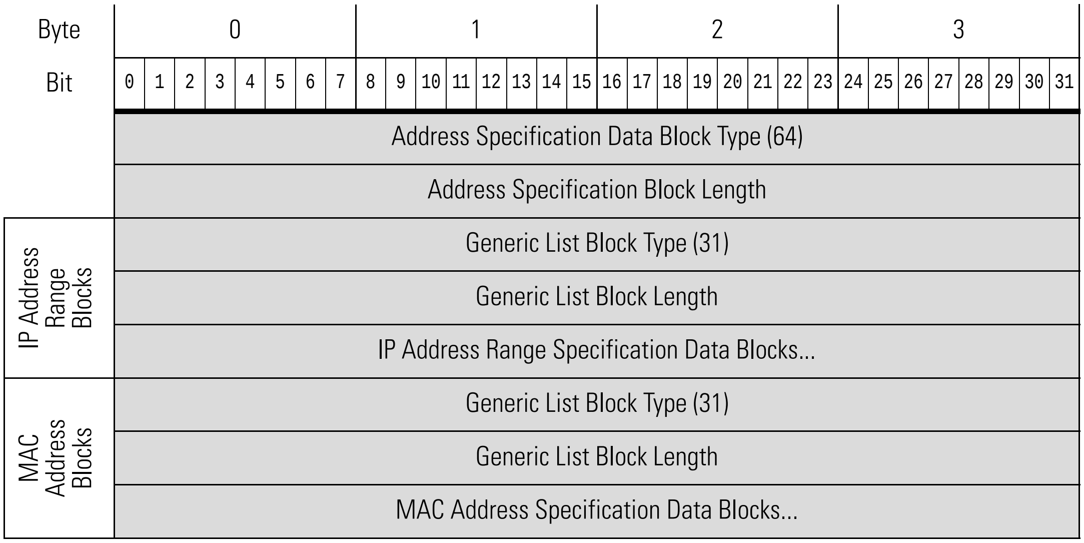
MAC Address Specification Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
MAC Address Specification Block Type	uint32	Initiates a MAC Address Specification data block. This value is always 63.
MAC Address Specification Block Length	uint32	Total number of bytes in the MAC Address Specification data block, including eight bytes for the MAC Address Specification block type and length fields, plus the number of bytes of MAC address specification data that follows.
MAC Address Blocks 1 - 6	uint8	The blocks of the MAC address in sequential order.

Address Specification Data Block

The Address Specification data block is used to contain lists of IP address range specifications and MAC address specifications. The Address Specification data block has a block type of 64 in the series 1 group of blocks.

The following diagram shows the basic structure of an Address Specification data block:


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Address Specification Data Block Type (64)
Address Specification Block Length
IP Address Range Blocks
Generic List Block Type (31)
Generic List Block Length
IP Address Range Specification Data Blocks...
MAC Address Blocks
Generic List Block Type (31)
Generic List Block Length
MAC Address Specification Data Blocks...

The Address Specification Data Block Fields table describes the fields of the Address Specification data block.

Address Specification Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
Address Specification Data Block Type	uint32	Initiates an Address Specification data block. This value is always 64.
Address Specification Block Length	uint32	Total number of bytes in the Address Specification data block, including eight bytes for the address specification block type and length fields, plus the number of bytes of address specification data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
IP Address Range Specification Data Blocks	variable	Encapsulated IP Address Range Specification data blocks up to the maximum number of bytes in the list block length. For more information, see IP Address Range Data Block for 5.2+ on page 270.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
MAC Address Specification Data Blocks	variable	Encapsulated MAC Address Specification data blocks up to the maximum number of bytes in the list block length. For more information, see MAC Address Specification Data Block on page 274.

Connection Chunk Data Block for 5.1.1+

The Connection Chunk data block conveys connection data. It stores connection log data that aggregates over a five-minute period. The Connection Chunk data block has a block type of 136 in the series 1 group of blocks. It supersedes block type 119. The following diagram shows the format of the Connection Chunk data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Connection Chunk Block Type (136)</td></tr>
<tr><td></td><td colspan="32">Connection Chunk Block Length</td></tr>
<tr><td></td><td colspan="32">Initiator IP Address</td></tr>
<tr><td></td><td colspan="32">Responder IP Address</td></tr>
<tr><td></td><td colspan="32">Start Time</td></tr>
<tr><td></td><td colspan="32">Application Protocol</td></tr>
<tr><td></td><td colspan="16">Responder Port</td><td colspan="8">Protocol</td><td colspan="8">Connection Type</td></tr>
<tr><td></td><td colspan="32">NetFlow Detector IP Address</td></tr>
<tr><td></td><td colspan="32">Packets Sent
Packets Sent, continued</td></tr>
<tr><td></td><td colspan="32">Packets Received
Packets Received, continued</td></tr>
<tr><td></td><td colspan="32">Bytes Sent
Bytes Sent, continued</td></tr>
<tr><td></td><td colspan="32">Bytes Received
Bytes Received, continued</td></tr>
<tr><td></td><td colspan="32">Connections</td></tr>
</table>

The Connection Chunk Data Block Fields table describes the components of the Connection Chunk data block.

Connection Chunk Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Chunk Block Type	uint32	Initiates a Connection Chunk data block. This value is always 136.
Connection Chunk Block Length	uint32	Total number of bytes in the Connection Chunk data block, including eight bytes for the connection chunk block type and length fields, plus the number of bytes in the connection chunk data that follows.
Initiator IP Address	uint8(4)	IP address of the initiator of this type of connection. This is used with the responder IP address to identify identical connections.
Responder IP Address	uint8(4)	IP address of the responder to this type of connection. This is used with the initiator IP address to identify identical connections.
Start Time	uint32	The starting time for the connection chunk.
Application Protocol	uint32	Identification number for the protocol used in the connection.
Responder Port	uint16	The port used by the responder in the connection chunk.
Protocol	uint8	The protocol for the packet containing the user information.
Connection Type	uint8	The type of connection.
NetFlow Detector IP Address	uint8[4]	IP address of the NetFlow device that detected the connection, in IP address octets.
Packets Sent	uint64	The number of packets sent in the connection chunk.
Packets Received	uint64	The number of packets received in the connection chunk.

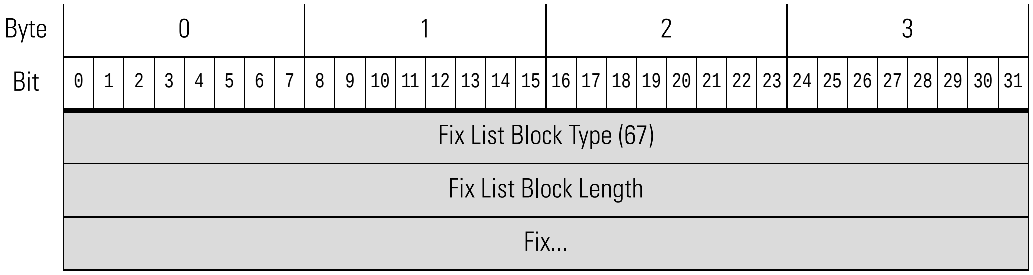
Connection Chunk Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Bytes Sent	uint64	The number of bytes sent in the connection chunk.
Bytes Received	uint64	The number of bytes received in the connection chunk.
Connections	uint32	The number of connections over a five-minute period.

Fix List Data Block

The Fix List data block conveys a fix that applies to a host. A Fix List data block for each fix applied to the affected host is included in a User Product data block. The Fix List data block has a block type of 67 in the series 1 group of blocks.

The following diagram shows the format of the Fix List data block:



The Fix List Data Block Fields table describes the components of the Fix List data block.

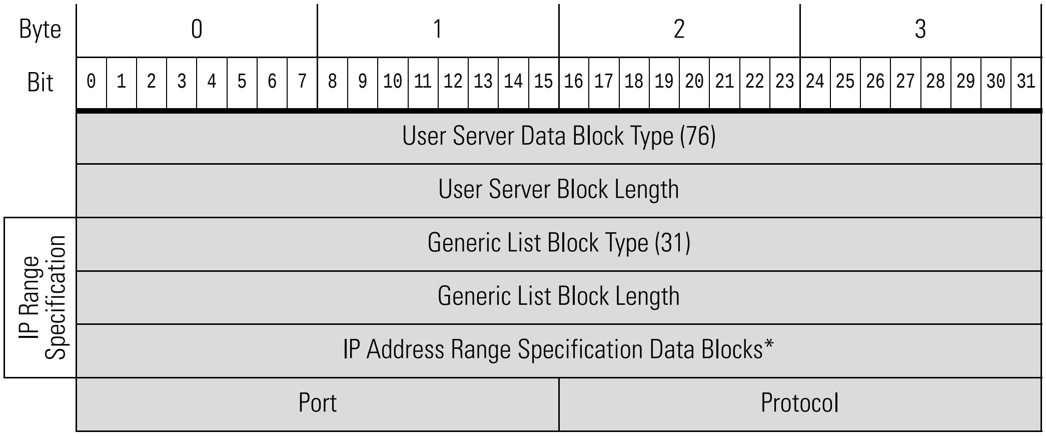
Fix List Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Fix List Block Type	uint32	Initiates a Fix List data block. This value is always 67.
Fix List Block Length	uint32	Total number of bytes in the Fix List data block, including eight bytes for the Fix List block type and length fields, plus the number of bytes of fix identification data that follows.
Fix ID	uint32	The identification number for the fix.

User Server Data Block

The User Server data block contains server details from a user input event. The User Server data block has a block type of 76 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Server data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	User Server Data Block Type (76)			
	User Server Block Length			
IP Range Specification	Generic List Block Type (31)			
	Generic List Block Length			
	IP Address Range Specification Data Blocks*			
	Port		Protocol	

The User Server Data Block Fields table describes the fields of the User Server data block.

User Server Data Block Fields

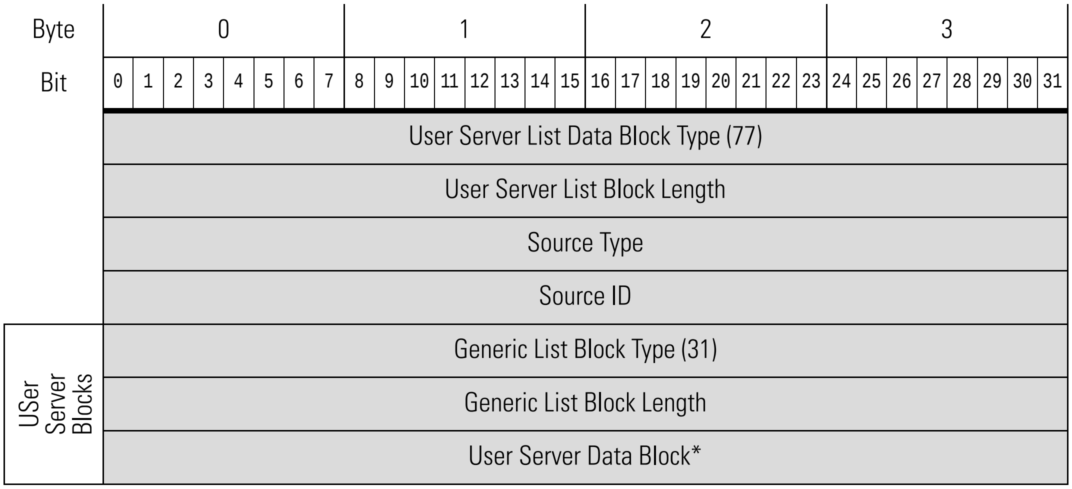
Field	Number of Bytes	Description
User Server Data Block Type	uint32	Initiates a User Server data block. This value is always 76.
User Server Block Length	uint32	Total number of bytes in the User Server data block, including eight bytes for the user server block type and length fields, plus the number of bytes of user server data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.

User Server Data Block Fields (Continued)

FIELD	NUMBER OF BYTES	DESCRIPTION
IP Address Range Specification Data Blocks	variable	Encapsulated IP Address Range Specification data blocks up to the maximum number of bytes in the list block length.
Port	uint16	Port used by the server.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP

User Server List Data Block

The User Server List data block contains a list of server data blocks from a user input event. The User Server List data block has a block type of 77 in the series 1 group of blocks. The following diagram shows the basic structure of a User Server List data block:



The User Server List Data Block Fields table describes the fields of the User Server List data block.

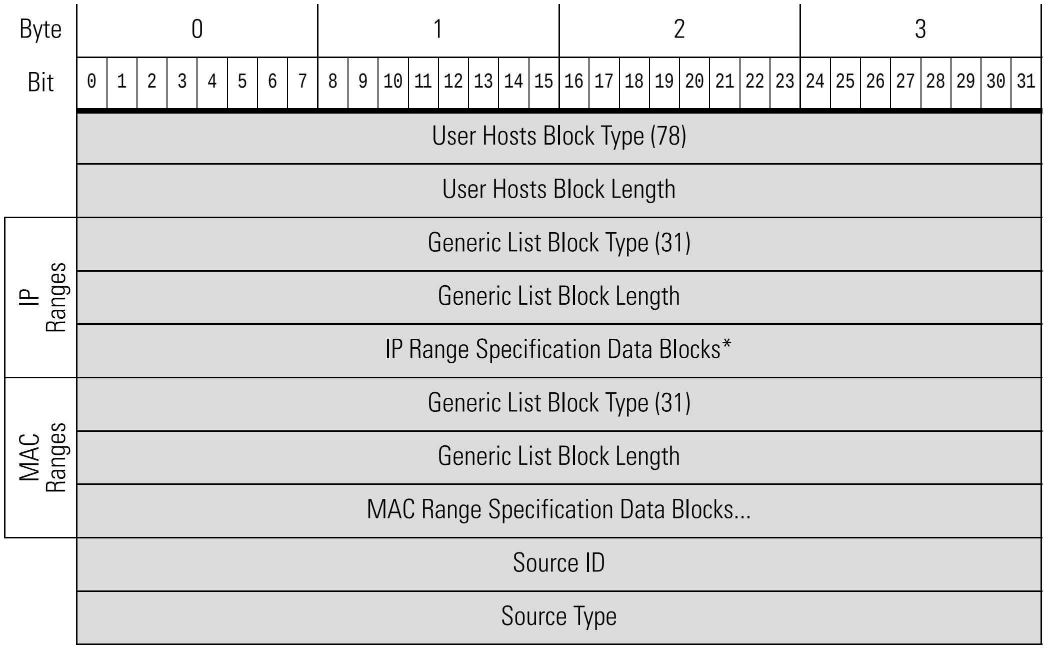
User Server List Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
User Server List Data Block Type	uint32	Initiates a User Server List data block. This value is always 77.
User Server List Block Length	uint32	Total number of bytes in the User Server List data block, including eight bytes for the user server list block type and length fields, plus the number of bytes of user server list data that follows.
Source Type	uint32	Number that maps to the type of data source: • `0` if the server data was detected by RNA • `1` if the server data was provided by a user • `2` if the server data was detected by a third-party scanner • `3` if the server data was provided by a command line tool such as nmimport.pl or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the server data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Server Data Blocks	variable	Encapsulated User Server data blocks up to the maximum number of bytes in the list block length.

User Hosts Data Block 4.7+

The User Hosts data block is used in User Add and Delete Host Messages on page 217 to contain information about host ranges and user and source identity from a user host input event. The User Hosts data block has a block type of 78 in the series 1 group of blocks.

The following diagram shows the basic structure of a User Hosts data block:


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
User Hosts Block Type (78)
User Hosts Block Length
IP Ranges
Generic List Block Type (31)
Generic List Block Length
IP Range Specification Data Blocks*
MAC Ranges
Generic List Block Type (31)
Generic List Block Length
MAC Range Specification Data Blocks...
Source ID
Source Type

The User Hosts Data Block Fields table describes the fields of the User Hosts data block.

User Hosts Data Block Fields

Field	Number of Bytes	Description
User Hosts Block Type	uint32	Initiates a User Hosts data block. This value is always 78.
User Hosts Block Length	uint32	Total number of bytes in the User Hosts data block, including eight bytes for the user hosts block type and length fields, plus the number of bytes of user hosts data that follows.

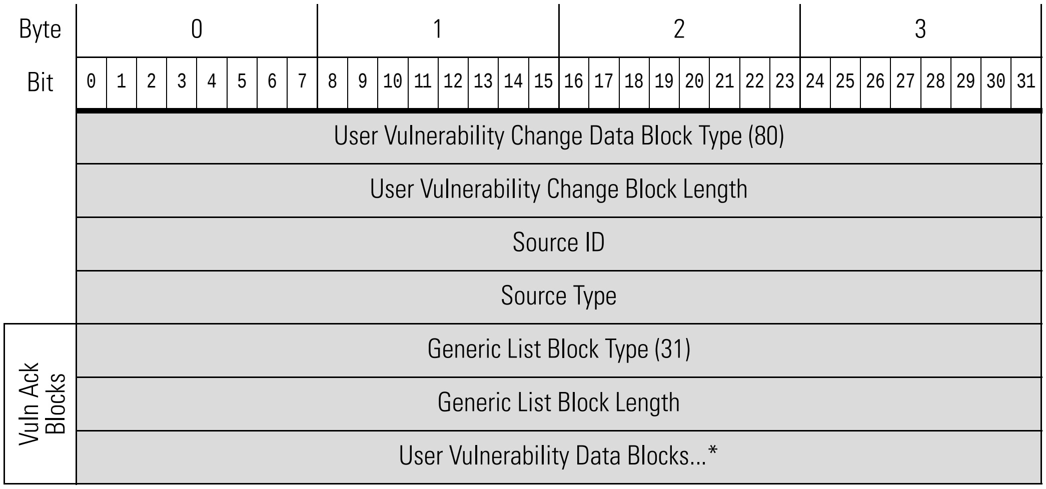
User Hosts Data Block Fields (Continued)

Field	Number of Bytes	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising MAC Range Specification data blocks conveying MAC address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated MAC Range Specification data blocks.
MAC Range Specification Data Blocks *	variable	MAC Range Specification data blocks containing information about the MAC address ranges for the user input. See MAC Address Specification Data Block on page 274 for a description of this data block.
Source ID	uint32	Identification number that maps to the source that added or updated the hostdata. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • `0` if the host data was detected by RNA • `1` if the host data was provided by a user • `2` if the host data was detected by a third-party scanner • `3` if the host data was provided by a command line tool such as nmimport.pl or the Host Input API client

User Vulnerability Change Data Block 4.7+

The User Vulnerability Change data block contains a list of deactivated vulnerabilities for the host, the identification number for the user who deactivated the vulnerabilities, information about the source that supplied the vulnerability changes, and the criticality value. The User Vulnerability Change data block has a block type of 80 in the series 1 group of blocks. Changes from the previous User Vulnerability Change data block include a new source type field and the use of the Generic list data block instead of the List data block to store vulnerability deactivations. This data block is used in user vulnerability change messages as documented in User Set Vulnerabilities Messages for Version 4.6.1+ on page 216.

The following diagram shows the basic structure of a User Vulnerability Change data block:



	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		User Vulnerability Change Data Block Type (80)			
		User Vulnerability Change Block Length			
		Source ID			
		Source Type			
Vuln Ack Blocks		Generic List Block Type (31)			
		Generic List Block Length			
		User Vulnerability Data Blocks...*			

The User Vulnerability Change Data Block Fields table describes the fields of the Generic List data block.

User Vulnerability Change Data Block Fields

Field	Number of Bytes	Description
User Vulnerability Change Data Block Type	uint32	Initiates a User Vulnerability Change data block. This value is always 80.
User Vulnerability Change Block Length	uint32	Total number of bytes in the User Vulnerability Change data block, including eight bytes for the host vulnerability block type and length fields, plus the number of bytes of host vulnerability data that follows.
Source ID	uint32	Identification number that maps to the source that updated or added the host vulnerability change value. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • 0 if the host vulnerability data was detected by RNA • 1 if the host vulnerability data was provided by a user • 2 if the host vulnerability data was detected by a third-party scanner • 3 if the host vulnerability data was provided by a command line tool such as nmimport.pl or the Host Input API client
Type	uint32	Type of vulnerability.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.

User Vulnerability Change Data Block Fields (Continued)

Field	Number of Bytes	Description
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Vulnerability Data Blocks	variable	Encapsulated User Vulnerability data blocks up to the maximum number of bytes in the list block length. For more information, see User Vulnerability Data Block 4.7 - 4.10.x on page 563 or User Vulnerability Data Block 5.0+ on page 336.

User Criticality Change Data Block 4.7+

The User Criticality data block is used to contain a list of IP address range specifications for hosts where the host criticality changed, the identification number for the user who updated the criticality value, information about the source that supplied the criticality value, and the criticality value. The User Criticality data block has a block type of 81 in the series 1 group of blocks. Changes from the previous User Criticality data block include a new source type field and the use of the Generic list data block instead of the List data block to store IP addresses.

The User Criticality data block is used in user set host criticality messages as documented in User Set Host Criticality Messages on page 218.

The following diagram shows the basic structure of a User Criticality data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	User Criticality Data Block Type (81)																															
	User Criticality Block Length																															

IP Address Range Blocks	Generic List Block Type (31)
	Generic List Block Length
	IP Address Range Specification Data Blocks...
	Source ID
	Source Type
	Criticality Value...

The User Criticality Data Block Fields table describes the fields of the User Criticality data block.

User Criticality Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
User Criticality Data Block Type	uint32	Initiates a User Criticality data block. This value is always 81.
User Criticality Block Length	uint32	Total number of bytes in the User Criticality data block, including eight bytes for the user criticality block type and length fields, plus the number of bytes of user criticality data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
IP Address Range Specification Data Blocks	variable	Encapsulated IP Address Range Specification data blocks up to the maximum number of bytes in the list block length.
Source ID	uint32	Identification number that maps to the source that updated or added the user criticality value. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.

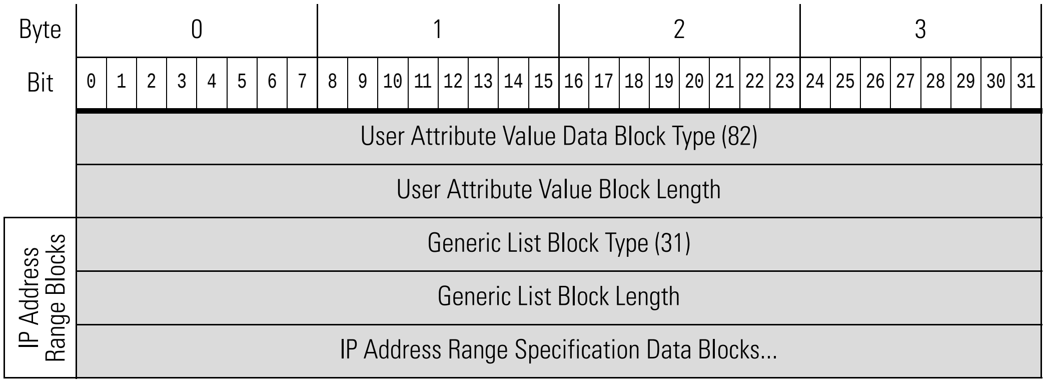
User Criticality Data Block Fields (Continued)

Field	Number of Bytes	Description
Source Type	uint32	Number that maps to the type of data source: • `0` if the user criticality value was provided by RNA • `1` if the user criticality value was provided by a user • `2` if the user criticality value was provided by a third-party scanner • `3` if the user criticality value was provided by a command line tool such as nmimport.pl or the Host Input API client
Criticality Value	uint32	User criticality value.

User Attribute Value Data Block 4.7+

The User Attribute Value data block contains a list of IP address ranges that indicate the hosts where the attribute value has changed, together with the identification number for the user who added the attribute value, information about the source that supplied the attribute value, and the BLOB data block containing the attribute value. The User Attribute Value data block has a block type of 82 in the series 1 group of blocks. Changes from the previous User Attribute Value data block include a new source type field and the use of the Generic list data block instead of the List data block to store IP addresses.

The following diagram shows the structure of a User Attribute Value data block:



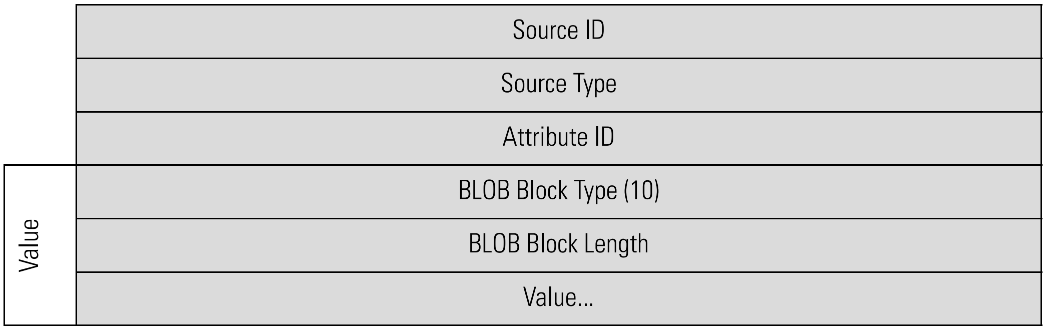
User Attribute Value Data Block Type (82)

User Attribute Value Block Length

Generic List Block Type (31)

Generic List Block Length

IP Address Range Specification Data Blocks...



The User Attribute Value Data Block Fields table describes the fields of the User Attribute Value data block.

User Attribute Value Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
User Attribute Value Data Block Type	uint32	Initiates a User Attribute Value data block. This value is always 82.
User Attribute Value Block Length	uint32	Total number of bytes in the Attribute Value data block, including eight bytes for the user attribute value block type and length fields, plus the number of bytes of user attribute value data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
IP Address Range Specification Data Blocks	variable	IP Address Range Specification data blocks (each with a start IP address and end IP address) up to the maximum number of bytes in the list block length.
Source ID	uint32	Identification number that maps to the source that added or updated the attribute data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.

User Attribute Value Data Block Fields (Continued)

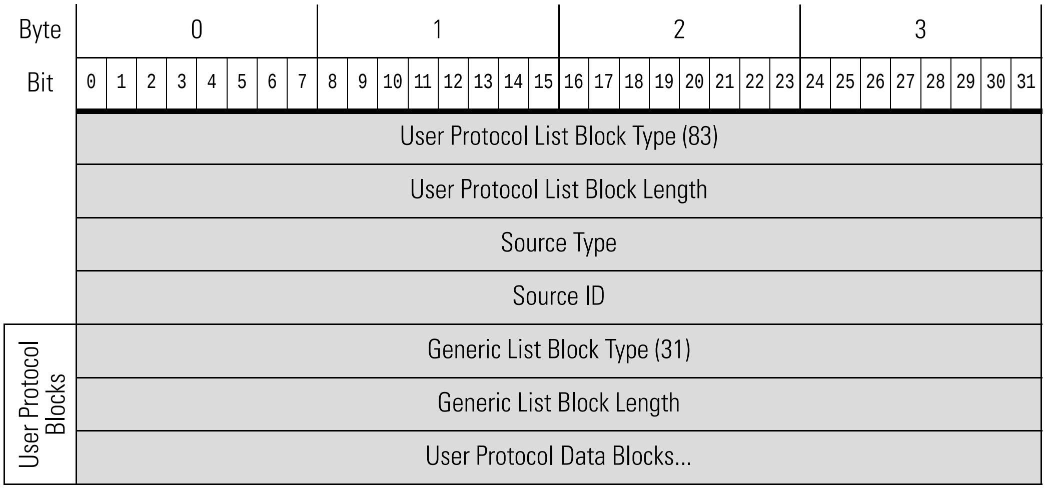
Field	Number of Bytes	Description
Source Type	uint32	Number that maps to the type of data source: • 0 if the user attribute value was provided by RNA • 1 if the user attribute value was provided by a user • 2 if the user attribute value was provided by a third-party scanner • 3 if the user attribute value was provided by a command line tool such as nmimport.pl or the Host Input API client
Attribute ID	uint32	Identification number of the updated attribute.
BLOB Block Type	uint32	Initiates a BLOB data block. This value is always 10.
BLOB Block Length	uint32	Number of bytes in the BLOB data block, including eight bytes for the BLOB block type and length fields, plus the length of the binary data that follows.
Value	variable	Contains the user attribute value, in binary format.

User Protocol List Data Block 4.7+

The User Protocol List data block is used to contain information about the source of the protocol data, the identification number for the user who added the data, and the lists of user protocol data blocks. The User Protocol List data block has a block type of 83 in the series 1 group of blocks. For more information on User Protocol data blocks, see User Protocol Data Block on page 265.

The User Protocol List data block is used in user protocol messages, as documented in User Protocol Messages on page 220.

The following diagram shows the basic structure of a User Protocol List data block:


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
User Protocol List Block Type (83)
User Protocol List Block Length
Source Type
Source ID
User Protocol Blocks
Generic List Block Type (31)
Generic List Block Length
User Protocol Data Blocks...

The User Protocol List Data Block Fields table describes the fields of the Generic List data block.

User Protocol List Data Block Fields

Field	Number of Bytes	Description
User Protocol List Block Type	uint32	Initiates a User Protocol List data block. This value is always 83.
User Protocol List Block Length	uint32	Total number of bytes in the User Protocol List data block, including eight bytes for the user protocol list block type and length fields, plus the number of bytes of user protocol list data that follows.
Source Type	uint32	Number that maps to the type of data source: • `0` if the protocol data was provided by RNA • `1` if the protocol data was provided by a user • `2` if the protocol data was provided by a third-party scanner • `3` if the protocol data was provided by a command line tool such as nmimport.pl or the Host Input API client

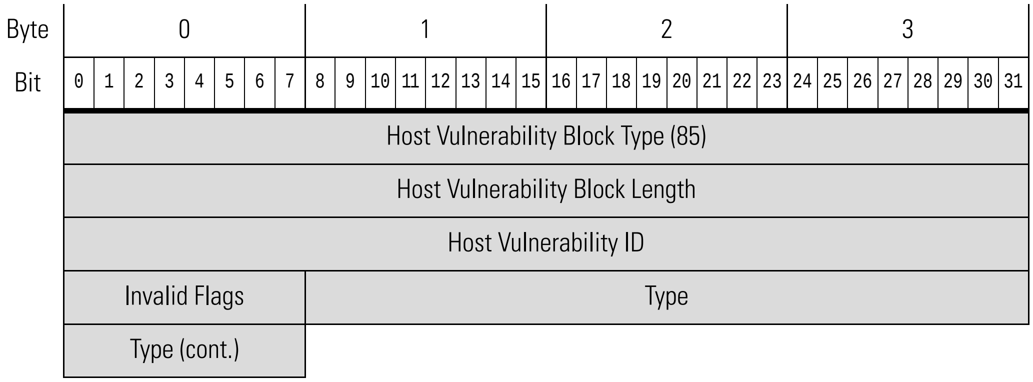
User Protocol List Data Block Fields (Continued)

FIELD	NUMBER OF BYTES	DESCRIPTION
Source ID	uint32	Identification number that maps to the source of the affected protocols. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
User Protocol Data Blocks	variable	Encapsulated User Protocol data blocks up to the maximum number of bytes in the list block length.

Host Vulnerability Data Block 4.9.0+

The Host Vulnerability data block conveys vulnerabilities that apply to a host. Each Host Vulnerability data block describes one vulnerability for a host in an event. Host Vulnerability data blocks appear in the Full Host Profile, Full Host Server, and Full Sub-Server data blocks. The Host Vulnerability data block has a block type of 85 in the series 1 group of blocks.

The following diagram shows the format of the Host Vulnerability data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Host Vulnerability Block Type (85)			
	Host Vulnerability Block Length			
	Host Vulnerability ID			
	Invalid Flags	Type		
	Type (cont.)			

The Host Vulnerability Data Block Fields table describes the components of the Host Vulnerability data block.

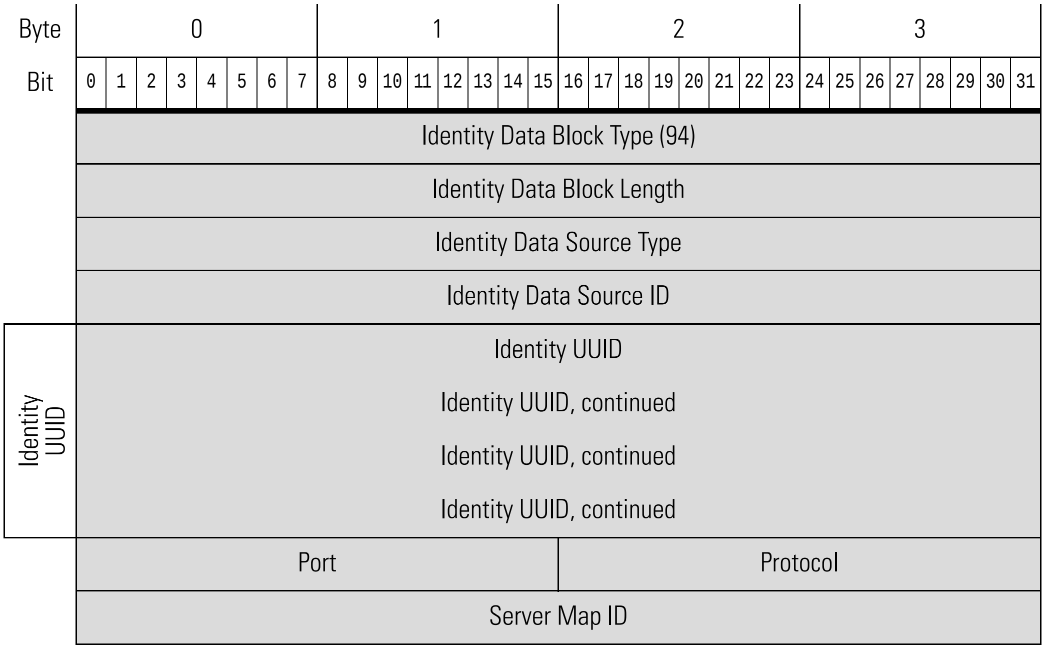
Host Vulnerability Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Host Vulnerability Block Type	uint32	Initiates an Host Vulnerability data block. This value is always 85.
Host Vulnerability Block Length	uint32	Total number of bytes in the Host Vulnerability data block, including eight bytes for the host vulnerability block type and length fields, plus the number of bytes of host vulnerability data that follows.
Host Vulnerability ID	uint32	The identification number for the vulnerability.
Invalid Flags	uint8	A value indicating whether the vulnerability is valid for the host.
Type	uint32	The type of vulnerability.

Identity Data Block

The identity data block has a block type of 94 in the series 1 group of blocks. Identity data blocks are used in identity conflict and identity timeout messages, which indicate when the identities of an operating system or server fingerprint source conflict or time out. The data block describes reported identities that have been identified as being in conflict with active source identities (user, scanner, or application). For more information, see Identity Conflict and Identity Timeout System Messages on page 222.

The following diagram shows the format of an identity data block for 4.9+.



	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		Identity Data Block Type (94)			
		Identity Data Block Length			
		Identity Data Source Type			
		Identity Data Source ID			
Identity UUID		Identity UUID			
		Identity UUID, continued			
		Identity UUID, continued			
		Identity UUID, continued			
		Port		Protocol	
		Server Map ID			

The Identity Data Block Fields table describes the fields of the Sourcefire identity data block.

Identity Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Identity Data Block Type	uint32	Initiates the Identity data block. This value is always 94.
Identity Data Block Length	uint32	Number of bytes in the Identity data block. This value should always be 40: sixteen bytes for the data block type and length fields and the source type and ID fields, sixteen bytes for the fingerprint UUID value, two bytes for the port, two bytes for the protocol, and four bytes for the SM ID.

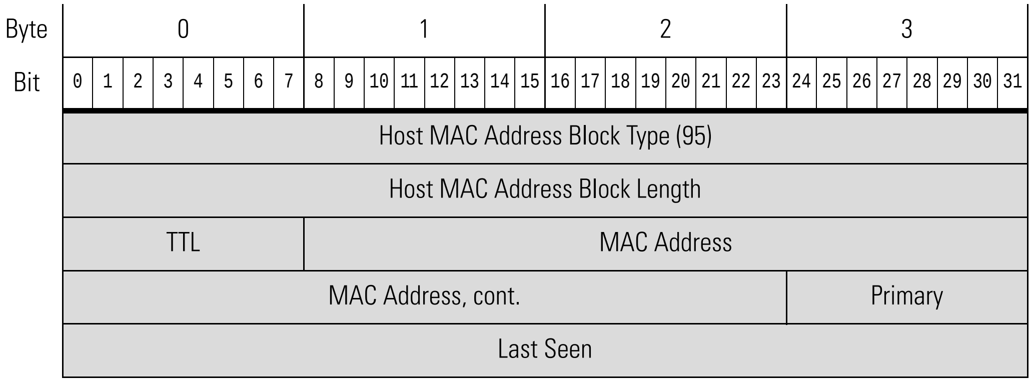
Identity Data Block Fields (Continued)

Field	Data Type	Description
Identity Data Source Type	uint32	Number that maps to the type of data source: • `0` if the fingerprint data was provided by RNA • `1` if the fingerprint data was provided by a user • `2` if the fingerprint data was provided by a third-party scanner • `3` if the fingerprint data was provided by a command line tool such as nmimport.pl or the Host Input API client
Identity Data Source ID	uint32	Identification number that maps to the source of the fingerprint data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
UUID	uint8[16]	If the identity is an operating system identity, the identification number, in octets, that acts as a unique identifier for the fingerprint.
Port	uint16	If the identity is a server identity, indicates the port used by the packet containing the server data.
Protocol	uint16	If the identity is a server identity, indicates the IANA number of the network protocol or Ethertype used by the packet containing the server data. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
Server Map ID	uint32	If the identity is a server identity, indicates the server map ID, representing the combination of ID, vendor, and version for the server.

Host MAC Address 4.9+

The host MAC address data block has a block type of 95 in the series 1 group of blocks. The block includes the time-to-live value for the host data, as well as the MAC address, the primary subnet of the host, and the last seen value for the host.

The following diagram shows the format of a host MAC address data block in 4.9+.



The Host MAC Address Data Block Fields table describes the fields of the Host MAC Address data block.

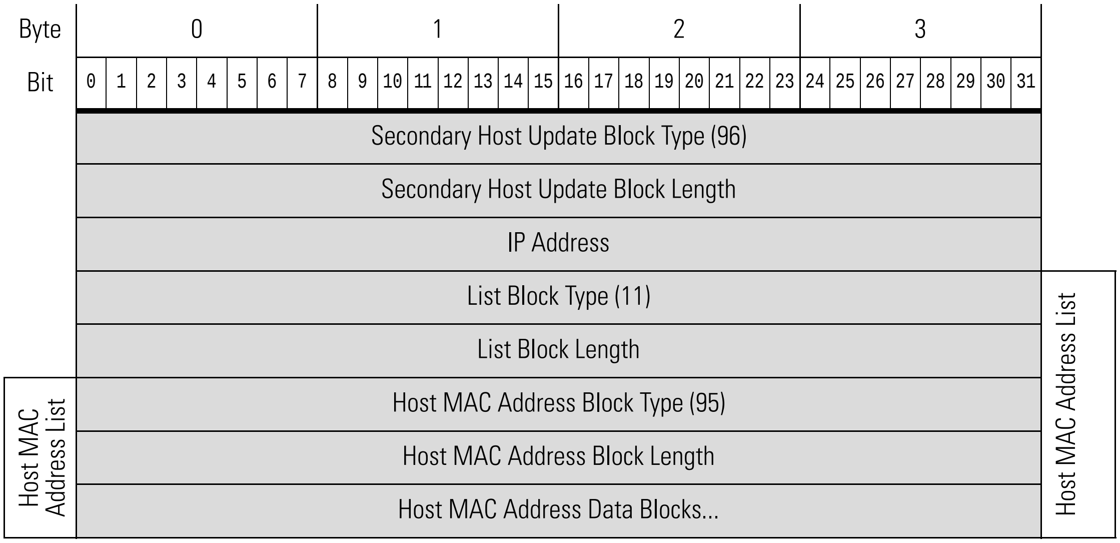
Host MAC Address Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Host MAC Address Data Block Type	uint32	Initiates the Host MAC Address data block. This value is always 95.
Host MAC Address Data Block Length	uint32	Number of bytes in the Host MAC Address data block. This value should always be 20: eight bytes for the data block type and length fields, one byte for the TTL value, 6 bytes for the MAC address, one byte for the primary subnet, and four bytes for the last seen value.
TTL	uint8	Indicates the difference between the TTL value in the packet used to fingerprint the host.
MAC Address	uint8 [6]	Indicates the MAC address of the host.
Primary	uint8	Indicates the primary subnet of the host.
Last Seen	uint32	Indicates when the host was last seen in traffic.

Secondary Host Update

The Secondary Host Update data block contains information for a host sent as a secondary host update from a device monitoring a subnet other than that where the host resides. It is used within Change Secondary Update events (event type 1001, subtype 31). The Secondary Host Update data block has a block type of 96 in the series 1 group of blocks.

The following diagram shows the format of a Secondary Host Update data block:



The Secondary Host Update Data Block Fields table describes the fields of the Secondary Host Update data block.

Secondary Host Update Data Block Fields

Field	Data Type	Description
Secondary Host Update Block Type	uint32	Initiates a Secondary Host Update data block. This value is always 96.
Secondary Host Update Block Length	uint32	Number of bytes in the Secondary Host Update data block, including eight bytes for the secondary host update block type and length fields, plus the number of bytes of secondary host update data that follows.
IP Address	uint8[4]	IP address of the host described in the update, in IP address octets.

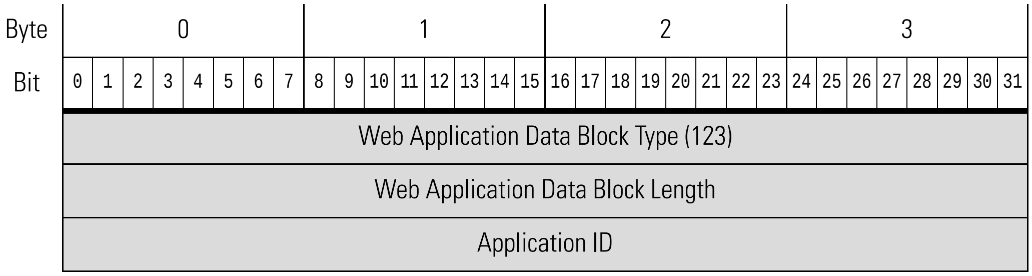
Secondary Host Update Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Type	uint32	Initiates a List data block comprising Host MAC Address data blocks conveying host MAC address data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Host MAC Address data blocks. This field is followed by zero or more Host MAC Address data blocks.
Host MAC Address Block Type	uint32	Initiates a Host MAC Address data block describing the secondary host. This value is always 95.
Host MAC Address Data Block Length	uint32	Number of bytes in the Host MAC Address data block. This value should always be 20: eight bytes for the data block type and length fields, one byte for the TTL value, 6 bytes for the MAC address, one byte for the primary subnet, and four bytes for the last seen value.
Host MAC Address Data Blocks	string	Information related to MAC addresses of hosts in the update.

Web Application Data Block for 5.0+

The Web Application data block for 5.0+ has a block type of 123 in the series 1 group of blocks. The data block describes the web application from detected HTTP client requests.

The following diagram shows the format of a Web Application data block in 5.0+.



The Web Application Data Block Fields table describes the fields of the Web Application data block.

Web Application Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Web Application Data Block Type	uint32	Initiates the Web Application data block. This value is always 123.
Web Application Data Block Length	uint32	Number of bytes in the Web Application data block, including eight bytes for the Web Application data block type and length, plus the number of bytes in the application ID field that follows.
Application ID	uint32	Application ID of the web application.

Connection Statistics Data Block 5.3+

The connection statistics data block is used in connection data messages. Changes to the connection data block between versions 5.2.x and 5.3 include the addition of new fields for NetFlow information. The connection statistics data block for version 5.3+ has a block type of 152 in the series 1 group of blocks. It deprecates block type 144, Connection Statistics Data Block 5.2.x on page 602.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 5.3+:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Connection Data Block Type (152)</td></tr>
<tr><td></td><td colspan="32">Connection Data Block Length</td></tr>
<tr><td></td><td colspan="32">Device ID</td></tr>
<tr><td></td><td colspan="32">Ingress Zone</td></tr>
<tr><td></td><td colspan="32">Ingress Zone, continued</td></tr>
<tr><td></td><td colspan="32">Ingress Zone, continued</td></tr>
<tr><td></td><td colspan="32">Ingress Zone, continued</td></tr>
</table>

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Egress Zone																															
	Egress Zone, continued																															
	Egress Zone, continued																															
	Egress Zone, continued																															
	Ingress Interface																															
	Ingress Interface, continued																															
	Ingress Interface, continued																															
	Ingress Interface, continued																															
	Egress Interface																															
	Egress Interface, continued																															
	Egress Interface, continued																															
	Egress Interface, continued																															
	Initiator IP Address																															
	Initiator IP Address, continued																															
	Initiator IP Address, continued																															
	Initiator IP Address, continued																															
	Responder IP Address																															
	Responder IP Address, continued																															
	Responder IP Address, continued																															
	Responder IP Address, continued																															
	Policy Revision																															
	Policy Revision, continued																															
	Policy Revision, continued																															
	Policy Revision, continued																															
	Rule ID																															
	Rule Action																Rule Reason															

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Initiator Port		Responder Port	
	TCP Flags		Protocol	NetFlow Source
	NetFlow Source, continued			
	NetFlow Source, continued			
	NetFlow Source, continued			
	NetFlow Source, continued			Instance ID
	Instance ID, cont.	Connection Counter		First Pkt Time
	First Packet Timestamp, continued			Last Pkt Time
	Last Packet Timestamp, continued			Initiator Tx Packets
	Initiator Transmitted Packets, continued			
	Initiator Transmitted Packets, continued			Resp. Tx Packets
	Responder Transmitted Packets, continued			
	Responder Transmitted Packets, continued			Initiator Tx Bytes
	Initiator Transmitted Bytes, continued			
	Initiator Transmitted Bytes, continued			Resp. Tx Bytes
	Responder Transmitted Bytes, continued			
	Responder Transmitted Bytes, continued			User ID
	User ID, continued			Application Prot. ID
	Application Protocol ID, continued			URL Category
	URL Category, continued			URL Reputation
	URL Reputation, continued			Client App ID
	Client Application ID, continued			Web App ID
Client URL	Web Application ID, continued			Str. Block Type (0)
	String Block Type, continued			String Block Length
	String Block Length, continued			Client App. URL...

<table>
<tr><th>Byte</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td rowspan="3">NetBIOS Name</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">NetBIOS Name...</td></tr>
<tr><td rowspan="3">Client App Version</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">Client Application Version...</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 1</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 2</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 3</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 4</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 5</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 6</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 7</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 8</td></tr>
<tr><td></td><td colspan="8">Sec. Int. Src/Dst</td><td colspan="8">Sec. Int. Layer</td><td colspan="16">File Event Count</td></tr>
<tr><td></td><td colspan="16">Intrusion Event Count</td><td colspan="16">Initiator Country</td></tr>
<tr><td></td><td colspan="16">Responder Country</td><td colspan="16">IOC Number</td></tr>
<tr><td></td><td colspan="32">Source Autonomous System</td></tr>
<tr><td></td><td colspan="32">Destination Autonomous System</td></tr>
<tr><td></td><td colspan="16">SNMP In</td><td colspan="16">SNMP Out</td></tr>
<tr><td></td><td colspan="8">Source TOS</td><td colspan="8">Destination TOS</td><td colspan="8">Source Mask</td><td colspan="8">Destination Mask</td></tr>
</table>

The Connection Statistics Data Block 5.2+ Fields table describes the fields of the Connection Statistics data block for 5.1.1+.

Connection Statistics Data Block 5.2+ Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.2+. The value is always 144.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.

Connection Statistics Data Block 5.2+ Fields (Continued)

Field	Data Type	Description
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.

Connection Statistics Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.

Connection Statistics Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint 16	Code for the country of the responding host.

Connection Statistics Data Block 5.2+ Fields (Continued)

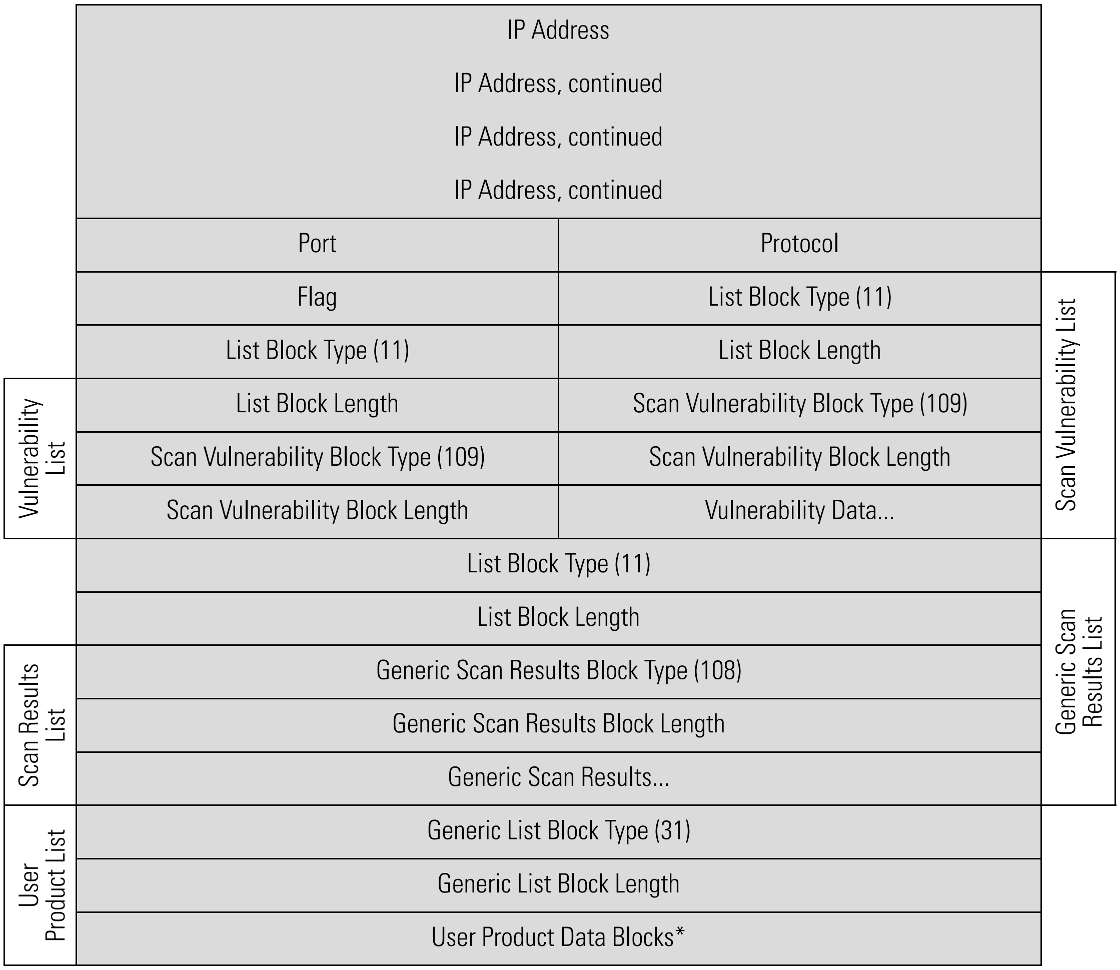
FIELD	DATA TYPE	DESCRIPTION
IOC Number	uint16	ID Number of the compromise associated with this event.
Source Autonomous System	uint32	Autonomous system number of the source, either origin or peer.
Destination Autonomous System	uint32	Autonomous system number of the destination, either origin or peer.
SNMP Input	uint16	SNMP index of the input interface.
SNMP Output	uint16	SNMP index of the output interface.
Source TOS	uint8	Type of Service byte setting for the incoming interface.
Destination TOS	uint8	Type of Service byte setting for the outgoing interface.
Source Mask	uint8	Source address prefix mask.
Destination Mask	uint8	Destination address prefix mask.

Scan Result Data Block 5.2+

The Scan Result data block describes a vulnerability and is used within Add Scan Result events (event type 1002, subtype 11). The Scan Result data block has a block type of 142 in the series 1 group of blocks. It supersedes block type 102. The IP address field was increased to 16 bytes for version 5.2.

The following diagram shows the format of a Scan Result data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Scan Result Block Type (142)</td></tr>
<tr><td></td><td colspan="32">Scan Result Block Length</td></tr>
<tr><td></td><td colspan="32">User ID</td></tr>
<tr><td></td><td colspan="32">Scan Type</td></tr>
</table>



	IP Address		
	IP Address, continued		
	IP Address, continued		
	IP Address, continued		
	Port	Protocol	
	Flag	List Block Type (11)	Scan Vulnerability List
	List Block Type (11)	List Block Length	
Vulnerability List	List Block Length	Scan Vulnerability Block Type (109)	
	Scan Vulnerability Block Type (109)	Scan Vulnerability Block Length	
	Scan Vulnerability Block Length	Vulnerability Data...	
	List Block Type (11)		Generic Scan Results List
	List Block Length		
Scan Results List	Generic Scan Results Block Type (108)		
	Generic Scan Results Block Length		
	Generic Scan Results...		
User Product List	Generic List Block Type (31)		
	Generic List Block Length		
	User Product Data Blocks*		

The Scan Result Data Block Fields table describes the fields of the Scan Result data block.

Scan Result Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Scan Result Block Type	uint32	Initiates a Scan Result data block. This value is always 142.
Scan Result Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.

Scan Result Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
User ID	uint32	Contains the user identification number for the user who imported the scan result or ran the scan that produced the scan result.
Scan Type	uint32	Indicates how the results were added to the system.
IP Address	uint8[16]	IP address of the host affected by the vulnerabilities in the result, in IP address octets.
Port	uint16	Port used by the sub-server affected by the vulnerabilities in the results.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
Flag	uint16	Reserved
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block describing a vulnerability detected during a scan. This value is always 109.

Scan Result Data Block Fields (Continued)

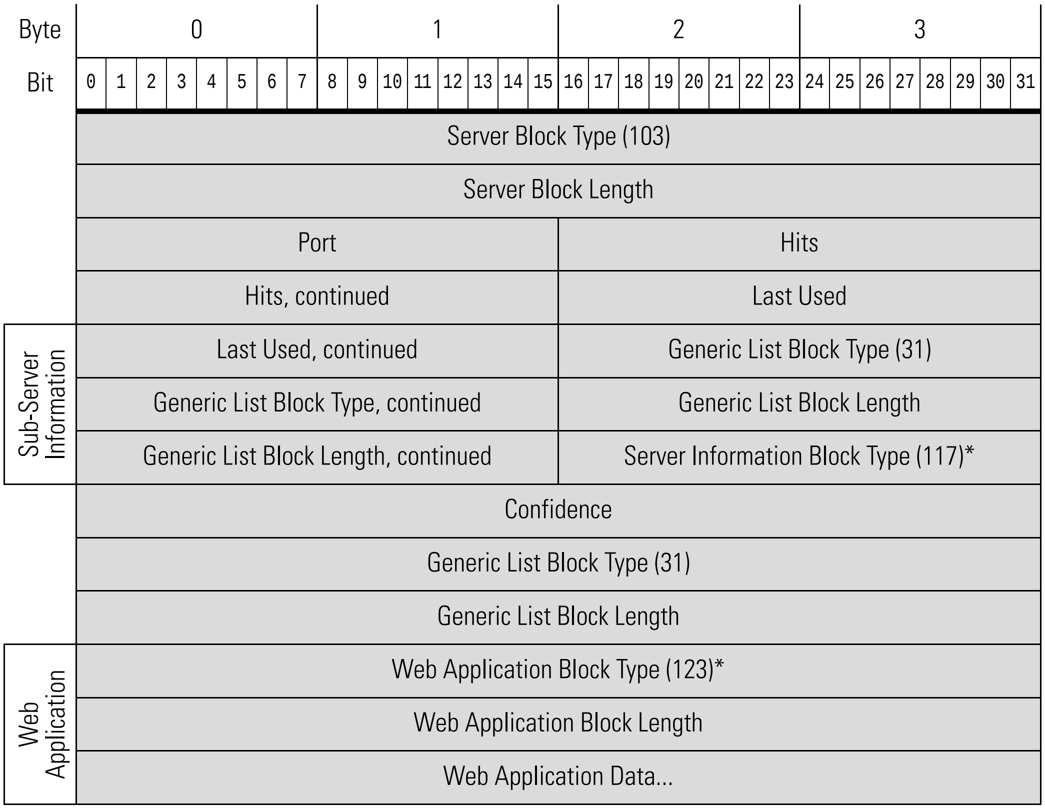
FIELD	DATA TYPE	DESCRIPTION
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes in the scan vulnerability data that follows.
Vulnerability Data	string	Information relating to each vulnerability.
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Generic Scan Results Block Type	uint32	Initiates a Generic Scan Results data block describing server and operating system data detected during a scan. This value is always 108.
Generic Scan Results Block Length	uint32	Number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes in the scan result data that follows.
Generic Scan Results Data	string	Information relating to each scan result.
Generic List Block Type	uint32	Initiates a Generic List data block comprising User Product data blocks conveying host input data from a third party application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated User Product data blocks.
User Product Data Blocks *	variable	User Product data blocks containing host input data. See User Product Data Block 5.1+ on page 353 for a description of this data block.

Host Server Data Block 4.10.0+

The Host Server data block conveys information about the detected servers on a host. It contains a block for each detected server, and also includes a list of web application data blocks for the web applications the server is running. Host Server data blocks are contained in messages for new and changed TCP and UDP servers. For more information, see Server Messages on page 206. The Host Server data block has a block type of 103 in the series 1 group of blocks.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Host Server data block:



<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Server Block Type (103)</td></tr>
<tr><td></td><td colspan="32">Server Block Length</td></tr>
<tr><td></td><td colspan="16">Port</td><td colspan="16">Hits</td></tr>
<tr><td></td><td colspan="16">Hits, continued</td><td colspan="16">Last Used</td></tr>
<tr><td rowspan="3">Sub-Server Information</td><td colspan="16">Last Used, continued</td><td colspan="16">Generic List Block Type (31)</td></tr>
<tr><td colspan="16">Generic List Block Type, continued</td><td colspan="16">Generic List Block Length</td></tr>
<tr><td colspan="16">Generic List Block Length, continued</td><td colspan="16">Server Information Block Type (117)*</td></tr>
<tr><td></td><td colspan="32">Confidence</td></tr>
<tr><td></td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td></td><td colspan="32">Generic List Block Length</td></tr>
<tr><td rowspan="3">Web Application</td><td colspan="32">Web Application Block Type (123)*</td></tr>
<tr><td colspan="32">Web Application Block Length</td></tr>
<tr><td colspan="32">Web Application Data...</td></tr>
</table>

The Host Server Data Block Fields table describes the fields of the Host Server data block.

Host Server Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Host Server Block Type	uint32	Initiates a Host Server data block. This value is always 103.
Host Server Block Length	uint32	Total number of bytes in the Host Server data block, including the eight bytes in the Host Server block type and length fields, plus the number of bytes of data that follows.
Port	uint16	Port number where the server runs.
Hits	uint32	Number of hits the server has received.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the server in use.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated sub-server information data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Server Information Data Blocks*	variable	Server information data blocks up to the maximum number of bytes in the list block length. For details, see Server Information Data Block for 4.10.x, 5.0 - 5.0.2 on page 319.
Confidence	uint32	Confidence percentage.
Generic List Block Type	uint32	Initiates a Generic data block. This value is always 31.

Host Server Data Block Fields (Continued)

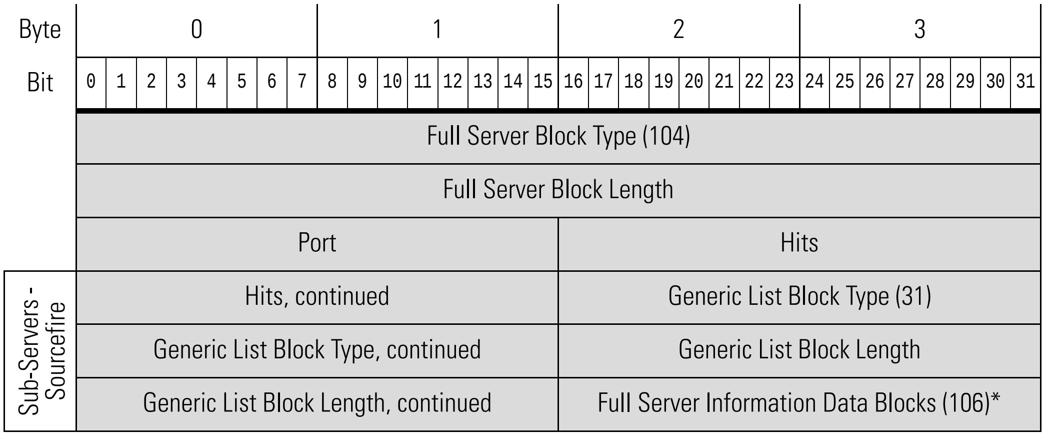
FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic block and encapsulated web application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated web application data blocks.
Web Application Data Blocks*	variable	Encapsulated web application data blocks up to the maximum number of bytes in the list block length. For details, see Web Application Data Block for 5.0+ on page 299.

Full Host Server Data Block 4.10.0+

The Full Host Server data block conveys information about a server, including the server port, the frequency of use and most recent update, confidence of data accuracy, and Sourcefire and third-party vulnerabilities related to that server for the host. The Full Host Server data block contains a Full Sub-Server Information data block for each sub-server on the server. Each Full Host Profile data block contains a Full Host Server data block for each TCP and UDP server on the host. The Full Host Server data block has a block type of 104 in the series 1 group of blocks.

IMPORTANT! An asterisk(*) next to a series 1 data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Server data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Server Block Type (104)			
	Full Server Block Length			
	Port		Hits	
Sub-Servers - Sourcefire	Hits, continued		Generic List Block Type (31)	
	Generic List Block Type, continued		Generic List Block Length	
	Generic List Block Length, continued		Full Server Information Data Blocks (106)*	

<table>
<tr><td></td><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td></td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="2" rowspan="3">Sub-Servers - User</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Full Server Information Data Block Type (106)*</td></tr>
<tr><td colspan="2" rowspan="3">Sub-Servers - Scanner</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Full Server Information Data Blocks (106)*</td></tr>
<tr><td colspan="2" rowspan="3">Sub-Servers - Application</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Full Server Information Data Blocks (106)*</td></tr>
<tr><td colspan="2"></td><td colspan="32">Confidence</td></tr>
<tr><td colspan="2" rowspan="3">Server Banner</td><td colspan="32">BLOB Block Type (10)</td></tr>
<tr><td colspan="32">BLOB Block Length</td></tr>
<tr><td colspan="32">Server Banner Data...</td></tr>
<tr><td colspan="2" rowspan="3">VDB Vulnerability</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">(VDB) Host Vulnerability Data Blocks (85)*</td></tr>
<tr><td colspan="2" rowspan="3">Third Pty/VDB Vulnerability</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">(Third Party/VDB) Host Vulnerability Data Blocks (85)*</td></tr>
<tr><td colspan="2" rowspan="3">Third Pty Host Vulnerability</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">(Third Party) Host Vulnerability Data Blocks (85)*</td></tr>
<tr><td colspan="2" rowspan="3">Web Application</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Web Application Data (123)*</td></tr>
</table>

The Full Server Data Block 4.10.0+ Fields table describes the components of the Full Server data block.

Full Server Data Block 4.10.0+ Fields

FIELD	DATA TYPE	DESCRIPTION
Full Server Block Type	uint32	Initiates a Full Server data block. This value is always 104.
Full Server Block Length	uint32	Total number of bytes in the Full Server data block, including eight bytes for the full server block type and length fields, plus the number of bytes of full server data that follows.
Port	uint16	Server port number.
Hits	uint32	Number of hits the server has received.
Generic List Block Type	uint32	Initiates a Generic List data block comprising data blocks of detected sub-server data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated sub-server information data blocks.
Sub-Server Information - Sourcefire Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers for a host server detected by Sourcefire. See Full Server Information Data Block on page 322 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising sub-server information data blocks conveying sub-server data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Sub-Server Information-User Added Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers on a host added by a user. See Full Server Information Data Block on page 322 for a description of this data block.

Full Server Data Block 4.10.0+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising sub-server information data blocks conveying sub-server data added by a scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated sub-server information data blocks.
Sub-Server Information-Scan Added Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers on a host added by a scanner. See Full Server Information Data Block on page 322 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising sub-server information data blocks conveying sub-server data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated sub-server information data blocks.
Sub-Server Information - Application Added Data Blocks *	variable	Full Server Information data blocks containing information about sub-servers on a host added by an application. See Full Server Information Data Block on page 322 for a description of this data block.
Confidence	uint32	Percentage of confidence of Sourcefire in correct identification of the full server data.
BLOB Block Type	uint32	Initiates a BLOB data block, which contains banner data. This value is always 10.
BLOB Block Length	uint32	Total number of bytes in the BLOB data block, including eight bytes for the block type and length fields, plus the number of bytes in the banner.
Server Banner Data	byte[*n*]	First *n* bytes of the packet involved in the server event, where *n* is equal to or less than 256.

Full Server Data Block 4.10.0+ Fields (Continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Sourcefire vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing Sourcefire information about host vulnerabilities in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party host vulnerability data sourced from a third party scanner and containing vulnerability information already cataloged in the Sourcefire VDB. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party host vulnerability data generated by a third party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third Party Scan Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing third party vulnerability data for vulnerabilities identified by a third party scanner but not cataloged in the Sourcefire VDB. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.

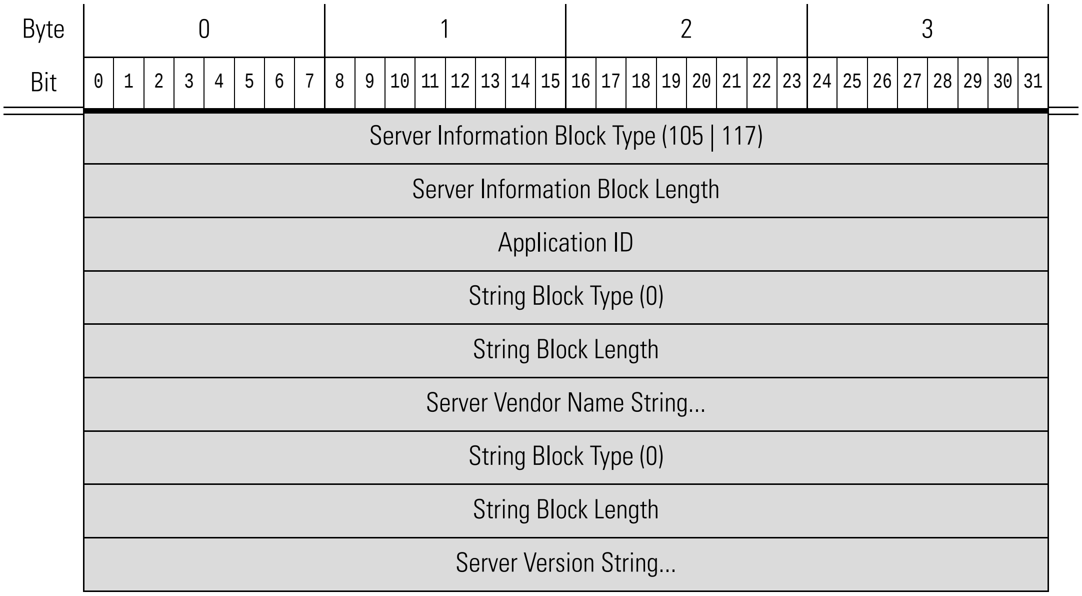
Full Server Data Block 4.10.0+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Web Application Data Blocks*	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length.

Server Information Data Block for 4.10.x, 5.0 - 5.0.2

The Server Information data block conveys information about a server, including the server ID, server vendor and version, and source information. The Server Information data block has a block type of 105 in the series 1 group of blocks for 4.10.x and a block type of 117 in the series 1 group of blocks for 5.0 - 5.0.2. Server information data blocks are conveyed in lists within Host Server blocks and Full Host server data blocks. For more information see Host Server Data Block 4.10.0+ on page 312 and Full Host Server Data Block 4.10.0+ on page 314.

The following diagram shows the format of the Server Information data block:



Server Information Block Type (105 | 117)

Server Information Block Length

Application ID

String Block Type (0)

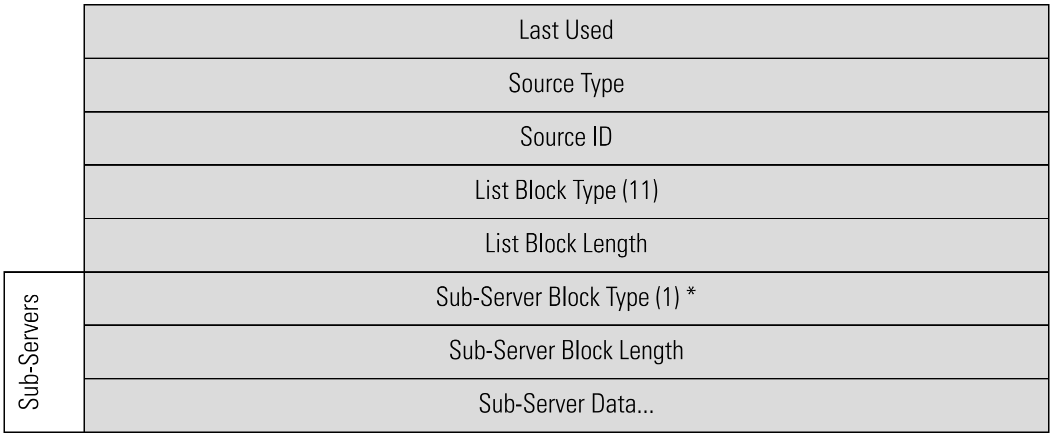
String Block Length

Server Vendor Name String...

String Block Type (0)

String Block Length

Server Version String...



Last Used

Source Type

Source ID

List Block Type (11)

List Block Length

Sub-Server Block Type (1) *

Sub-Server Block Length

Sub-Server Data...

The Server Information Data Block Fields table describes the components of the Server Information data block.

Server Information Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Server Information Block Type	uint32	Initiates a Server Information data block. The block type is 105 for 4.10.x and 117 for 5.0+.
Server Information Block Length	uint32	Total number of bytes in the Server Information data block, including eight bytes for the Server Information block type and length fields, four bytes for the server ID, eight bytes for the vendor name block type and length, another four for the vendor name, eight bytes for the version string block type and length, another four for the version string, and four bytes each for the last used, source type, and source ID fields.
Application ID	uint32	The application ID for the application protocol running on the detected server.
String Block Type	uint32	Initiates a String data block containing the server vendor’s name. This value is always 0.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the server vendor name.
Server Vendor Name	string	Name of the server vendor.

Server Information Data Block Fields (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block that contains the server version. This value is always 0.
String Block Length	uint32	Number of bytes in the server version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the server version.
Server Version	string	Server version.
Last Time Used	uint32	Indicates when the server information was last used in traffic.
Source Type	uint32	Number that maps to the type of data source: • `0` if the server data was provided by RNA • `1` if the server data was provided by a user • `2` if the server data was provided by a third-party scanner • `3` if the server data was provided by a command line tool such as nmimport.pl or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the server data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
List Block Type	uint32	Initiates a list of Sub-Server data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including eight bytes for the list block type and length fields, plus the number of bytes in the encapsulated Sub-Server data blocks that follow.
Sub-Server Block Type	uint32	Initiates the first Sub-Server data block. This data block can be followed by other Sub-Server data blocks up to the limit defined in the list block length field.

Server Information Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Sub-Server Block Length	uint32	Total number of bytes in each Sub-Server data block, including the eight bytes in the Sub-Server block type and length fields, plus the number of bytes of data that follows.
Sub-Server Data	variable	Sub-server data as documented in Sub-Server Data Block on page 241.

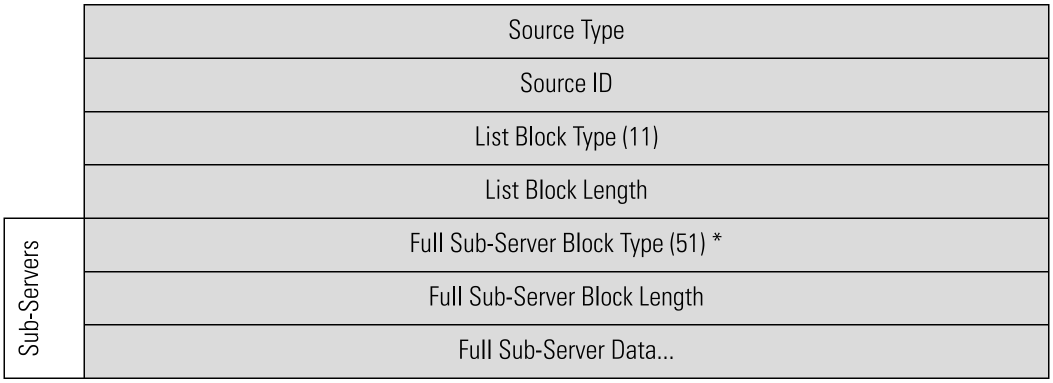
Full Server Information Data Block

The Full Server Information data block conveys information about a server detected on a host, including the server's application protocol, vendor, and version, and the list of its associated sub-servers. For each sub-server, information is included by a Full Sub-Server data block (see Full Sub-Server Data Block on page 255). The Full Server Information data block has a block type of 106 in the series 1 group of blocks.

IMPORTANT! An asterisk(*) next to a series 1 data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Server Information data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Server Block Type (106)			
	Full Server Block Length			
	Application Protocol ID			
Vendor	String Block Type (0)			
	String Block Length			
	Vendor Name String...			
Version	String Block Type (0)			
	String Block Length			
	Version String...			
	Last Used			



The Full Server Information Data Block Fields table describes the components of the Full Server Information data block.

Full Server Information Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Full Server Information Block Type	uint32	Initiates a Full Server Information data block. This value is always 106.
Full Server Information Block Length	uint32	Total number of bytes in the Full Server Information data block, including eight bytes for the full server block type and length fields, plus the number of bytes in the full server data that follows.
Application Protocol ID	uint32	The application ID of the application protocol running on the server.
String Block Type	uint32	Initiates a String data block containing the application protocol vendor's name. This value is always 0.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the vendor name.
Vendor Name	string	Name of the server vendor.
String Block Type	uint32	Initiates a String data block that contains the application protocol version. This value is always 0.

Full Server Information Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Version	string	The version of the server.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the server in use.
Source Type	uint32	Number that maps to the type of data source: • `0` if the server data was provided by RNA • `1` if the server data was provided by a user • `2` if the client data was provided by a third-party scanner • `3` if the server data was provided by a command line tool such as nmimport.pl or the Host Input API client
Source ID	uint32	Identification number that maps to the source of the server data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
List Block Type	uint32	Initiates a List data block comprising Full Server Information data blocks conveying sub-server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Full Sub-Server data blocks. This field is followed by zero or more Full Sub-Server data blocks.
Full Sub-Server Block Type	uint32	Initiates the first Full Sub-Server data block. This data block can be followed by other Full Sub-Server data blocks up to the limit defined in the list block length field.

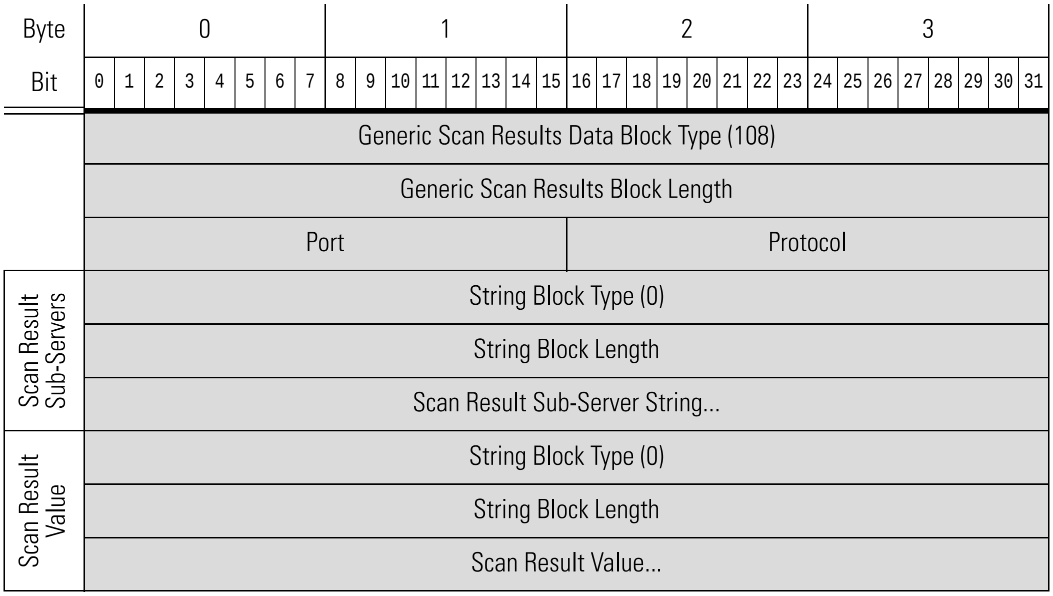
Full Server Information Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Full Sub-Server Block Length	uint32	Total number of bytes in each Full Sub-Server data block, including the eight bytes in the Full Sub-Server block type and length fields, plus the number of bytes of data that follows.
Full Sub-Server Data Blocks *	uint32	Full Sub-Server data blocks containing sub-servers for the server. See Full Sub-Server Data Block on page 255 for a description of this data block.

Generic Scan Results Data Block for 4.10.0+

The Generic Scan Results data block contains scan results and is used in the Scan Result Data Block 5.2+ on page 308. The Generic Scan Results data block has a block type of 108 in the series 1 group of blocks.

The following diagram shows the basic structure of a Generic Scan Results data block:



Scan Result Sub-Server	String Block Type (0)
	String Block Length
	Scan Result Sub-Server (unformatted) String...
Scan Result Value	String Block Type (0)
	String Block Length
	Scan Result Value...

The Generic Scan Result Data Block Fields table describes the fields of the Generic Scan Results data block.

Generic Scan Result Data Block Fields

Field	Number of Bytes	Description
Generic Scan Results Data Block Type	uint32	Initiates a Generic Scan Results data block. This value is always 108.
Generic Scan Results Block Length	uint32	Total number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes of scan results data that follows.
Port	uint16	Port used by the server affected by the vulnerabilities in the results.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
String Block Type	uint32	Initiates a String data block that contains the sub-server. This value is always 0.

Generic Scan Result Data Block Fields (Continued)

FIELD	NUMBER OF BYTES	DESCRIPTION
String Block Length	uint32	Number of bytes in the sub-server String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server.
Scan Result Sub-Server	string	Sub-server.
String Block Type	uint32	Initiates a String data block that contains the value. This value is always 0.
String Block Length	uint32	Number of bytes in the value String data block, including eight bytes for the block type and length fields, plus the number of bytes in the value.
Scan result value	string	Scan result value.
String Block Type	uint32	Initiates a String data block that contains the sub-server. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server.
Scan Result Sub-Server	string	Sub-server (unformatted).
String Block Type	uint32	Initiates a String data block that contains the value. This value is always 0.
String Block Length	uint32	Number of bytes in the value String data block, including eight bytes for the block type and length fields, plus the number of bytes in the value.
Scan Result Value	string	Scan result value (unformatted).

Scan Vulnerability Data Block for 4.10.0+

The Scan Vulnerability data block describes a vulnerability and is used within Scan Result data blocks, which in turn are used in Add Scan Result events (event type 1002, subtype 11). For more information, see Scan Result Data Block 5.2+ on page 308 and Add Scan Result Messages on page 221. The Scan Vulnerability data block has a block type of 109 in the series 1 group of blocks.

The following diagram shows the format of a Scan Vulnerability data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Scan Vulnerability Block Type (109)			
	Scan Vulnerability Block Length			
	Port		Protocol	
ID	String Block Type (0)			
	String Block Length			
	ID			
Name	String Block Type (0)			
	String Block Length			
	Vulnerability Name...			
Description	String Block Type (0)			
	String Block Length			
	Description...			
Name Clean	String Block Type (0)			
	String Block Length			
	Vulnerability Name Clean...			
Description Clean	String Block Type (0)			
	String Block Length			
	Description Clean...			

Bugtraq ID	List Block Type (11)
	List Block Length
	Integer Data Blocks (Bugtraq IDs)...
CVE ID	List Block Type (11)
	List Block Length
	CVE ID...

The Scan Vulnerability Data Block Fields table describes the fields of the Scan Vulnerability data block.

Scan Vulnerability Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block. This value is always 109.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
Port	uint16	Port used by the sub-server affected by the vulnerability.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
String Block Type	uint32	Initiates a String data block for the ID.

Scan Vulnerability Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the String data block for the ID, including eight bytes for the string block type and length, plus the number of bytes in the ID.
ID	string	The ID for the reported vulnerability as specified by the scan utility that detected it. For a vulnerability detected by a Qualys scan, for example, this field indicates the Qualys ID.
String Block Type	uint32	Initiates a String data block for the vulnerability name.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Name	string	Name of the vulnerability.
String Block Type	uint32	Initiates a String data block for the vulnerability description.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability description, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability description.
Description	string	Description of the vulnerability.
String Block Type	uint32	Initiates a String data block for the vulnerability name.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Name Clean	string	Name of the vulnerability (unformatted).
String Block Type	uint32	Initiates a String data block for the vulnerability description.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability description, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability description.

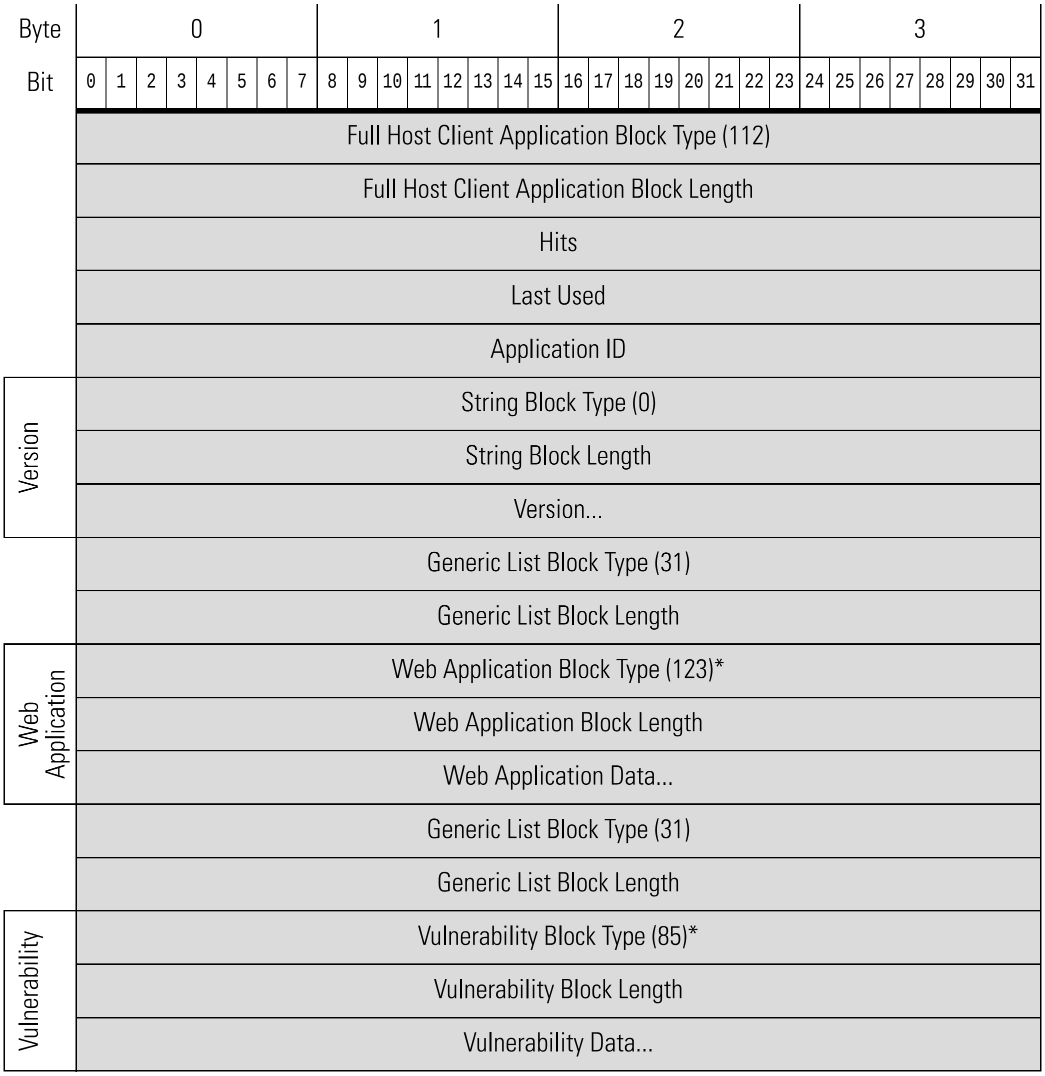
Scan Vulnerability Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Description Clean	string	Description of the vulnerability (unformatted).
List Block Type	uint32	Initiates a List data block for the list of Bugtraq identification numbers.
List Block Length	uint32	Number of bytes in the List data block for the list of Bugtraq identification numbers, including eight bytes for the string block type and length, plus the number of bytes in the Integer data blocks containing the Bugtraq IDs.
Bugtraq ID	string	Contains zero or more Integer (INT32) data blocks that form a list of Bugtraq identification numbers. For more information on these data blocks, see Integer (INT32) Data Block on page 244.
List Block Type	uint32	Initiates a List data block for the list of Common Vulnerability Exposure (CVE) identification numbers.
List Block Length	uint32	Number of bytes in the List data block for the CVE identification number, including eight bytes for the string block type and length, plus the number of bytes in the CVE identification number.
CVE ID	string	Contains zero or more String Information data blocks that form a list of CVE identification numbers. For more information on these data blocks, see String Information Data Block on page 249.

Full Host Client Application Data Block 5.0+

The Full Host Client Application data block for version 5.0+ describes a client application, plus an appended list of associated web applications and vulnerabilities. The Full Host Client Application data block is used within the Full Host Profile data block (type 111). It has a block type of 112 in the series 1 group of blocks.

The following diagram shows the basic structure of a Full Host Client Application data block for 5.0+:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Host Client Application Block Type (112)			
	Full Host Client Application Block Length			
	Hits			
	Last Used			
	Application ID			
Version	String Block Type (0)			
	String Block Length			
	Version...			
	Generic List Block Type (31)			
	Generic List Block Length			
Web Application	Web Application Block Type (123)*			
	Web Application Block Length			
	Web Application Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Vulnerability	Vulnerability Block Type (85)*			
	Vulnerability Block Length			
	Vulnerability Data...			

The Full Host Client Application Data Block 5.0+ Fields table describes the fields of the Full Host Client Application data block.

Full Host Client Application Data Block 5.0+ Fields

FIELD	DATA TYPE	DESCRIPTION
Full Host Client Application Block Type	uint32	Initiates a Full Host Client Application data block. This value is always 112.
Full Host Client Application Block Length	uint32	Number of bytes in the Full Host Client Application data block, including eight bytes for the client application block type and length, plus the number of bytes in the client application data that follows.
Hits	uint32	Number of times the system has detected the client application in use.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the client in use.
Application ID	uint32	Application ID of the detected client application, if applicable. For more information on client applications, see the *Sourcefire 3D System eStreamer Integration Guide*.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application name, including eight bytes for the string block type and length, plus the number of bytes in the client application version.
Version	string	Client application version.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and the encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.

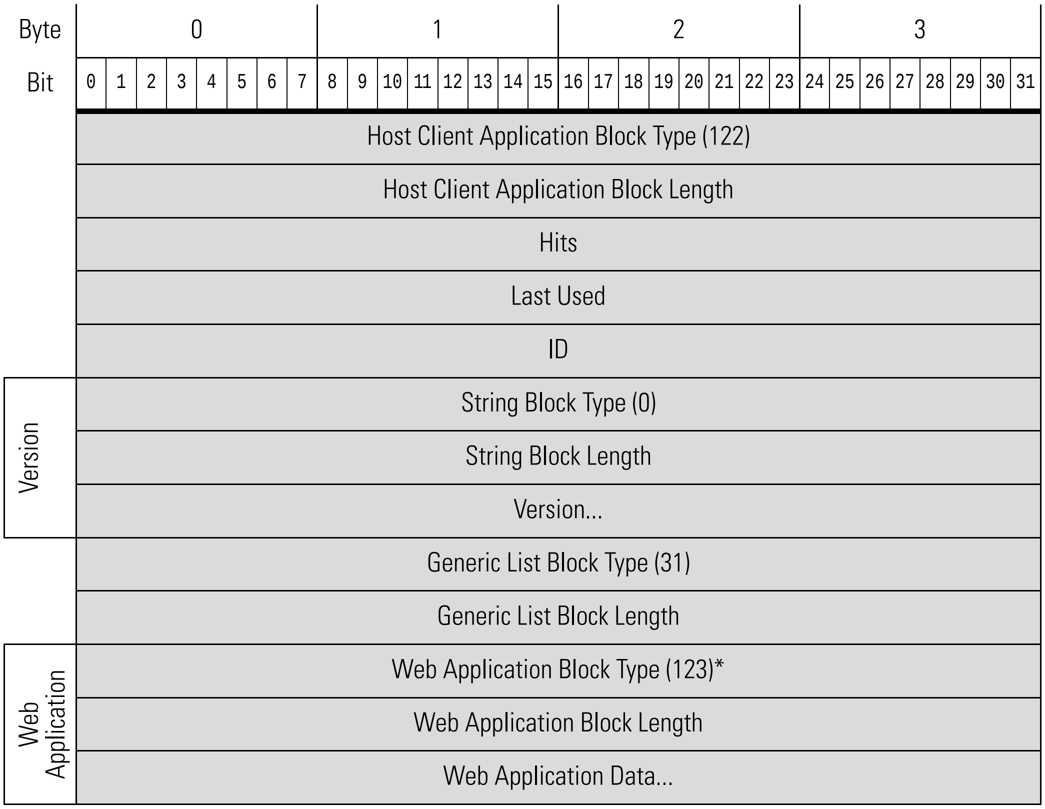
Full Host Client Application Data Block 5.0+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the generic list block length.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Vulnerability data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated Vulnerability data blocks.
Vulnerability Data Blocks	variable	Encapsulated Vulnerability data blocks up to the maximum number of bytes in the generic list block length.

Host Client Application Data Block for 5.0+

The Host Client Application data block for 5.0+ describes a client application and is used within New Client Application events (event type 1000, subtype 7), Client Application Timeout events (event type 1001, subtype 20), and Client Application Update events (event type 1001, subtype 32). The Host Client Application data block for 4.10.2+ has a block type of 122 in the series 1 group of blocks.

The following diagram shows the basic structure of a Host Client Application data block for 5.0+:



The Host Client Application Data Block Fields table describes the fields of the Host Client Application data block.

Host Client Application Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Client Application Block Type	uint32	Initiates a Host Client Application data block. This value is always 122.
Client Application Block Length	uint32	Number of bytes in the Client Application data block, including eight bytes for the client application block type and length, plus the number of bytes in the client application data that follows.

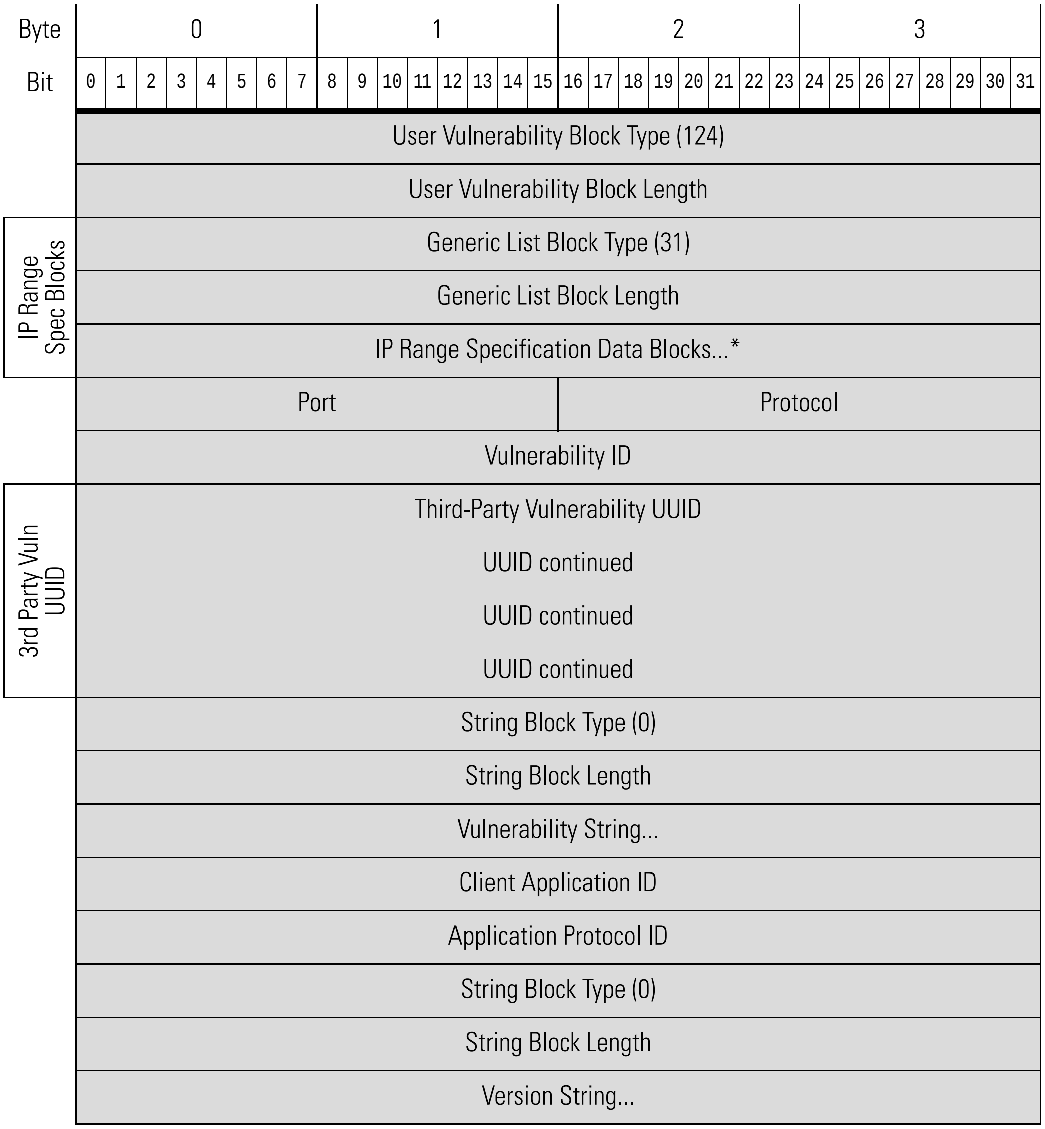
Host Client Application Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Hits	uint32	Number of times the system has detected the client application in use.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the client in use.
ID	uint32	Identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the client application version.
Version	string	Client application version.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length. See Web Application Data Block for 5.0+ on page 299 for information on the encapsulated data blocks (block type 123).

User Vulnerability Data Block 5.0+

The User Vulnerability data block describes a vulnerability and is used within User Vulnerability Change data blocks. These in turn are used in User Set Valid Vulnerabilities events and User Set Invalid Vulnerabilities events. The User Vulnerability data block for 5.0+ has a block type of 124 in the series 1 group of blocks. It supersedes block type 79. For more information on User Vulnerability Change data blocks, see User Vulnerability Change Data Block 4.7+ on page 285.

The following diagram shows the format of a User Vulnerability data block:



	Byte 0	Byte 1	Byte 2	Byte 3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	User Vulnerability Block Type (124)			
	User Vulnerability Block Length			
IP Range Spec Blocks	Generic List Block Type (31)			
	Generic List Block Length			
	IP Range Specification Data Blocks...*			
	Port		Protocol	
	Vulnerability ID			
3rd Party Vuln UUID	Third-Party Vulnerability UUID			
	UUID continued			
	UUID continued			
	UUID continued			
	String Block Type (0)			
	String Block Length			
	Vulnerability String...			
	Client Application ID			
	Application Protocol ID			
	String Block Type (0)			
	String Block Length			
	Version String...			

The User Vulnerability Data Block Fields table describes the fields of the User Vulnerability data block.

User Vulnerability Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
User Vulnerability Block Type	uint32	Initiates a User Vulnerability data block. This value is always 124.
User Vulnerability Block Length	uint32	Number of bytes in the User Vulnerability data block, including eight bytes for the user vulnerability block type and length fields, plus the number of bytes of user vulnerability data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP address ranges from user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Port	uint16	Port used by the server affected by the vulnerability. For client application vulnerabilities, the value is 0.
Protocol	uint16	IANA protocol number or Ethertype for the protocol used by the server affected by the vulnerability. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP For client application vulnerabilities, the value is `0`.

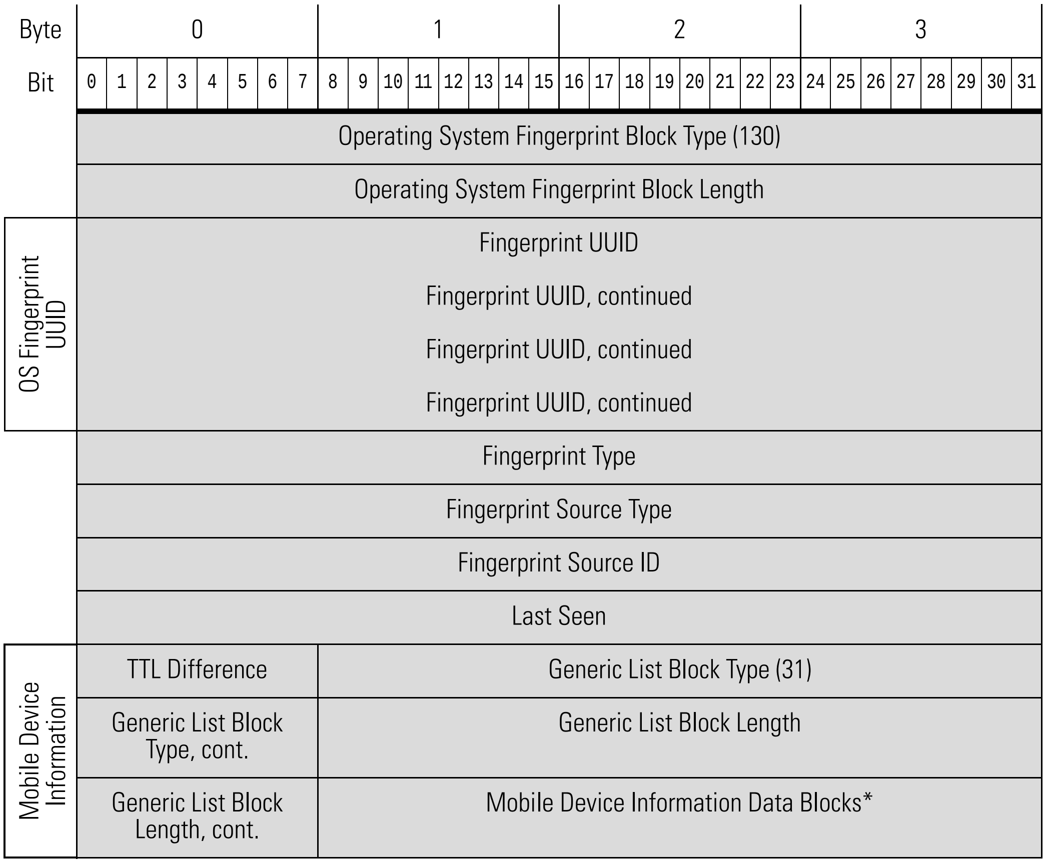
User Vulnerability Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Vulnerability ID	uint32	The Sourcefire vulnerability ID.
Third-Party Vulnerability UUID	uint8 [16]	A unique ID number for the third-party vulnerability, if one exists. Otherwise, the value is 0.
String Block Type	uint32	Initiates a String data block for the vulnerability name. The value is always 0.
String Block Length	uint32	The number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Vulnerability Name	string	The vulnerability name.
Client Application ID	uint32	The application ID of the client application. For server vulnerabilities, the value is 0.
Application Protocol ID	uint32	The application ID of the application protocol used by client application. For server vulnerabilities, the value is 0.
String Block Type	uint32	Initiates a String data block for the version string. The value is always 0.
String Block Length	uint32	The number of bytes in the String data block for the version, including eight bytes for the string block type and length, plus the number of bytes in the client application version string.
Version	string	The client application version. For server vulnerabilities, the value is 0.

Operating System Fingerprint Data Block 5.1+

The Operating System Fingerprint data block has a block type of 130 in the series 1 group of blocks. The block includes a fingerprint Universally Unique Identifier (UUID), as well as the fingerprint type, the fingerprint source type, and the fingerprint source ID.

The following diagram shows the format of an Operating System Fingerprint data block in 5.1+.



	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		Operating System Fingerprint Block Type (130)			
		Operating System Fingerprint Block Length			
OS Fingerprint UUID		Fingerprint UUID			
		Fingerprint UUID, continued			
		Fingerprint UUID, continued			
		Fingerprint UUID, continued			
		Fingerprint Type			
		Fingerprint Source Type			
		Fingerprint Source ID			
		Last Seen			
Mobile Device Information		TTL Difference	Generic List Block Type (31)		
		Generic List Block Type, cont.	Generic List Block Length		
		Generic List Block Length, cont.	Mobile Device Information Data Blocks*		

The Operating System Fingerprint Data Block Fields table describes the fields of the operating system fingerprint data block.

Operating System Fingerprint Data Block Fields

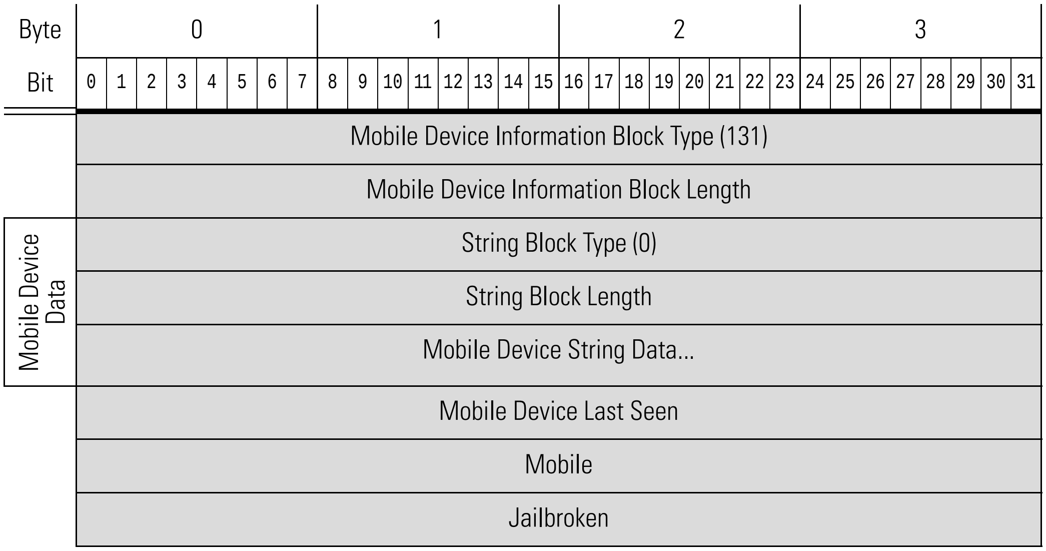
FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint Data Block Type	uint32	Initiates the operating system data block. This value is always 130.
Operating System Data Block Length	uint32	Number of bytes in the Operating System Fingerprint data block, including eight bytes for the Operating System Fingerprint Data Block block type and length, plus the number of bytes in the Operating System Fingerprint data that follows.
Fingerprint UUID	uint8[16]	Fingerprint identification number, in octets, that acts as a unique identifier for the operating system. The fingerprint UUID maps to the operating system name, vendor, and version in the vulnerability database (VDB).
Fingerprint Type	uint32	Indicates the type of fingerprint.
Fingerprint Source Type	uint32	Indicates the type (i.e., user or scanner) of the source that supplied the operating system fingerprint.
Fingerprint Source ID	uint32	Identification number that maps to the login name of the user that supplied the operating system fingerprint.
Last Seen	uint32	Indicates when the fingerprint was last seen in traffic.
TTL Difference	uint8	Indicates the difference between the TTL value in the fingerprint and the TTL value seen in the packet used to fingerprint the host.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.

Operating System Fingerprint Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Mobile Device Information Data Blocks	variable	Encapsulated Mobile Device Information data blocks up to the maximum number of bytes in the list block length. See Mobile Device Information Data Block for 5.1+ on page 342 for a description of this data block.

Mobile Device Information Data Block for 5.1+

The following diagram shows the format of a Mobile Device Information data block. The data block contains the last time the host was detected, mobile device information, and whether the mobile device is jailbroken. The Mobile Device Information data block has a block type of 131 in the series 1 group of blocks.



The describes the fields of the Mobile Device Information data block returned by 5.1+.

Mobile Device Information Data Block 5.1+ Fields

FIELD	DATA TYPE	DESCRIPTION
Mobile Device Information Block Type (131)	uint32	Initiates the operating system data block. This value is always 131.
Mobile Device Information Block Length	uint32	Number of bytes in the Mobile Device Information data block, including eight bytes for the Mobile Device Information Data Block block type and length, plus the number of bytes in the Mobile Device Information data that follows.
String Block Type	uint32	Initiates a string data block for the mobile device string. This value is set to 0 to indicate string data.
String Block Length	uint32	Indicates the number of bytes in the mobile device string data block, including eight bytes for the string block type and length fields, plus the number of bytes in the mobile device string data that follows.
Mobile Device String Data	Variable	Contains the mobile device hardware information of the host detected.
Mobile Device Last Seen	uint32	Contains the time stamp the mobile device was last seen.
Mobile	uint32	True-false flag indicating whether the host is a mobile device.
Jailbroken	uint32	True-false flag indicating whether the host is a mobile device that is jailbroken.

Host Profile Data Block for 5.2+

The following diagram shows the format of a Host Profile data block. The data block also does not include a host criticality value, but does include a VLAN presence indicator. In addition, a data block can convey a NetBIOS name for the host. The Host Profile data block has a block type of 139 in the series 1 group of blocks. The data block now supports IPv6 addresses, and client application data blocks have been added.

IMPORTANT! An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.

<table>
<tr><td>Byte</td><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td></tr>
<tr><td></td><td colspan="4">Host Profile Block Type (139)</td></tr>
<tr><td></td><td colspan="4">Host Profile Block Length</td></tr>
<tr><td></td><td colspan="4">IP Address</td></tr>
<tr><td></td><td colspan="4">IP Address, continued</td></tr>
<tr><td></td><td colspan="4">IP Address, continued</td></tr>
<tr><td></td><td colspan="4">IP Address, continued</td></tr>
<tr><td rowspan="3">Server Fingerprints</td><td>Hops</td><td>Primary/Secondary</td><td colspan="2">Generic List Block Type (31)</td></tr>
<tr><td colspan="2">Generic List Block Type, continued</td><td colspan="2">Generic List Block Length</td></tr>
<tr><td colspan="2">Generic List Block Length, continued</td><td colspan="2">Server Fingerprint Data Blocks*</td></tr>
<tr><td rowspan="3">Client Fingerprints</td><td colspan="4">Generic List Block Type (31)</td></tr>
<tr><td colspan="4">Generic List Block Length</td></tr>
<tr><td colspan="4">Client Fingerprint Data Blocks*</td></tr>
<tr><td rowspan="3">SMB Fingerprints</td><td colspan="4">Generic List Block Type (31)</td></tr>
<tr><td colspan="4">Generic List Block Length</td></tr>
<tr><td colspan="4">SMB Fingerprint Data Blocks*</td></tr>
<tr><td rowspan="3">DHCP Fingerprints</td><td colspan="4">Generic List Block Type (31)</td></tr>
<tr><td colspan="4">Generic List Block Length</td></tr>
<tr><td colspan="4">DHCP Fingerprint Data Blocks*</td></tr>
</table>

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
Mobile Device Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	Mobile Device Fingerprint Data Blocks*				
IPv6 Sever Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	Ipv6 Server Fingerprint Data Blocks*				
IPv6 Client Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	IPv6 Client Fingerprint Data Blocks*				
IPv6 DHCP Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	IPv6 DHCP Fingerprint Data Blocks*				
User Agent Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	User Agent Fingerprint Data Blocks*				
TCP Server Block*	List Block Type (11)				List of TCP Servers
	List Block Length				
	TCP Server Data Blocks				
UDP Server Block*	List Block Type (11)				List of UDP Servers
	List Block Length				
	UDP Server Data Blocks				

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
Network Protocol Block*	List Block Type (11)				List of Network Protocols
	List Block Length				
	Network Protocol Data Blocks				
Transport Protocol Block*	List Block Type (11)				List of Transport Protocols
	List Block Length				
	Transport Protocol Data Blocks				
MAC Address Block*	List Block Type (11)				List of MAC Addresses
	List Block Length				
	Host MAC Address Data Blocks				
	Host Last Seen				
	Host Type				
	Mobile	Jailbroken	VLAN Presence	VLAN ID	
Client App Data	VLAN ID, cont.	VLAN Type	VLAN Priority	Generic List Block Type (31)	List of Client Applications
	Generic List Block Type (31), cont.			Generic List Block Length	
	Generic List Block Length, cont.			Client Application Data Blocks	
NetBIOS Name	String Block Type (0)				
	String Block Length				
	NetBIOS String Data...				

The Host Profile Data Block 5.2+ Fields table describes the fields of the host profile data block returned by 5.2+.

Host Profile Data Block 5.2+ Fields

FIELD	DATA TYPE	DESCRIPTION
Host Profile Block Type	uint32	Initiates the Host Profile data block for 5.2+. This value is always 139.
Host Profile Block Length	uint32	Number of bytes in the Host Profile data block, including eight bytes for the host profile block type and length fields, plus the number of bytes included in the host profile data that follows.
IP Address	uint8(16)	IP Address of the host. This can be IPv4 or IPv6.
Hops	uint8	Number of hops from the host to the Device.
Primary/ Secondary	uint8	Indicates whether the host is in the primary or secondary network of the device that detected it: • 0 — host is in the primary network. • 1 — host is in the secondary network.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Host Profile Data Block 5.2+ Fields (Continued)

Field	Data Type	Description
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an SMB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (SMB Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an SMB fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a mobile device fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Host Profile Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint Mobile) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a mobile device fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Server) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Client) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Host Profile Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint (IPv6 DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a user agent fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Agent Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a user agent fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying TCP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
TCP Server Data Blocks	variable	Host server data blocks describing a TCP server. See Host Server Data Block 4.10.0+ on page 312 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying UDP server data. This value is always 11.

Host Profile Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
UDP Server Data Blocks	uint32	Host server data blocks describing a UDP server. See Host Server Data Block 4.10.0+ on page 312 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more Protocol data blocks.
Network Protocol Data Blocks	uint32	Protocol data blocks describing a network protocol. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more transport protocol data blocks.
Transport Protocol Data Blocks	uint32	Protocol data blocks describing a transport protocol. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising MAC Address data blocks. This value is always 11.

Host Profile Data Block 5.2+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated MAC Address data blocks.
Host MAC Address Data Blocks	uint32	Host MAC Address data blocks describing a host MAC address. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Host Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates the host type. The following values may appear: • 0 — host • 1 — router • 2 — bridge • 3 — NAT device • 4 — LB (load balancer)
Mobile	uint8	True-false flag indicating whether the host is a mobile device.
Jailbroken	uint8	True-false flag indicating whether the host is a mobile device that is also jailbroken.
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
String Block Type	uint32	Initiates a String data block for the host client application data. This value is always 112.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the host client application data.

Host Profile Data Block 5.2+ Fields (Continued)

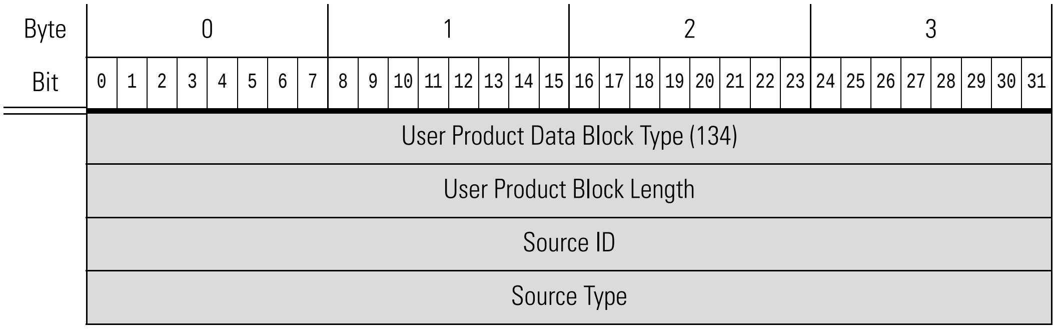
FIELD	DATA TYPE	DESCRIPTION
Host Client Application Data Blocks	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+ on page 331 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.

User Product Data Block 5.1+

The User Product data block conveys host input data imported from a third party application, including third party application string mappings. This data block is used in Scan Result Data Block 5.2+ on page 308 and User Server and Operating System Messages on page 219. The User Product data block has a block type of 65 in the series 1 group of blocks for versions up to 4.7-4.10.1, a block type of 118 for 4.10.2-5.0.x, and a block type of 134 in the series 1 group of blocks for 5.1+. Block types 65 and 118 have the same structure.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the User Product data block:



IP Address Ranges	Generic List Block Type (31)	
	Generic List Block Length	
	IP Range Specification Data Blocks*	
	Port	Protocol
	Drop User Product	
Custom Vendor String	String Block Type (0)	
	String Block Length	
	Custom Vendor String...	
Custom Product String	String Block Type (0)	
	String Block Length	
	Custom Product String...	
Custom Version String	String Block Type (0)	
	String Block Length	
	Custom Version String...	
	Software ID	
	Server ID	
	Vendor ID	
	Product ID	
Major Version String	String Block Type (0)	
	String Block Length	
	Major Version String...	
Minor Version String	String Block Type (0)	
	String Block Length	
	Minor Version String...	
Revision String	String Block Type (0)	
	String Block Length	
	Revision String...	

To Major String	String Block Type (0)
	String Block Length
	To Major Version String...
To Minor String	String Block Type (0)
	String Block Length
	To Minor Version String...
To Revision String	String Block Type (0)
	String Block Length
	To Revision String...
Build String	String Block Type (0)
	String Block Length
	Build String...
Patch String	String Block Type (0)
	String Block Length
	Patch String...
Extension String	String Block Type (0)
	String Block Length
	Extension String...
OS UUID	Operating System UUID
	Operating System UUID cont.
	Operating System UUID cont.
	Operating System UUID cont.

<table>
<tr><td rowspan="3">Device String String</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">Device String String...</td></tr>
<tr><td rowspan="4">List of Fixes</td><td>Mobile</td><td>Jailbroken</td><td colspan="2">Generic List Block Type (31)</td></tr>
<tr><td colspan="2">Generic List Block Type (31) cont.</td><td colspan="2">Generic List Block Length</td></tr>
<tr><td colspan="2">Generic List Block Length cont.</td><td colspan="2">Fix List Data Blocks*</td></tr>
<tr><td colspan="2">Fix List Data Blocks* cont.</td><td colspan="2"></td></tr>
</table>

The User Product Data Block Fields table describes the components of the User Product data block.

User Product Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
User Product Data Block Type	uint32	Initiates a User Product data block. This value is 134 for 5.1+.
User Product Block Length	uint32	Total number of bytes in the User Product data block, including eight bytes for the user product block type and length fields, plus the number of bytes in the user product data that follows.
Source ID	uint32	Identification number that maps to the source that imported the data. Depending on the source type, this may map to RNA, a user, a scanner, or a third-party application.
Source Type	uint32	Number that maps to the type of data source: • `0` if the data was provided by RNA • `1` if the data was provided by a user • `2` if the data was provided by a third-party scanner • `3` if the data was provided by a command line tool such as nmimport.pl or the Host Input API client
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.

User Product Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Port	uint16	Port specified by the user.
Protocol	uint16	IANA protocol number or Ethertype. This is handled differently for Transport and Network layer protocols. Transport layer protocols are identified by the IANA protocol number. For example: • 6 — TCP • 17 — UDP Network layer protocols are identified by the decimal form of the IEEE Registration Authority Ethertype. For example: • 2048 — IP
Drop User Product	uint32	Indicates whether the user OS definition was deleted from the host: • `0` — No • `1` — Yes
String Block Type	uint32	Initiates a String data block containing the custom vendor name specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom vendor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the vendor name.
Custom Vendor Name	string	The custom vendor name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom product name specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom product String data block, including eight bytes for the block type and length fields, plus the number of bytes in the product name.

User Product Data Block Fields (Continued)

Field	Data Type	Description
Custom Product Name	string	The custom product name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom version specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Custom Version	string	The custom version specified in the user input.
Software ID	uint32	The identifier for a specific revision of a server or operating system in the Sourcefire database.
Server ID	uint32	The Sourcefire application identifier for the application protocol on the host server specified in user input.
Vendor ID	uint32	The identifier for the vendor of a third party operating system specified when the third party operating system is mapped to a Sourcefire 3D operating system definition.
Product ID	uint32	The product identification string of a third party operating system string specified when the third party operating system string is mapped to a Sourcefire 3D operating system definition.
String Block Type	uint32	Initiates a String data block containing the major version number of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Major Version	string	Major version of the Sourcefire 3D operating system definition that a third party operating system string is mapped to.

User Product Data Block Fields (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the minor version number of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Minor Version	string	Minor version number of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the revision number of the Sourcefire operating system definition that a third party operating system string in the user input is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
Revision	string	Revision number of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last major version of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Major	string	Last version number in a range of major version numbers of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.

User Product Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block containing the last minor version of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Minor	string	Last version number in a range of minor version numbers of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the Last revision number of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
To Revision	string	Last revision number in a range of revision numbers of the Sourcefire 3D operating system definitions that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the build number of the Sourcefire 3D operating system that the third party operating system string is mapped. This value is always 0.
String Block Length	uint32	Number of bytes in the build String data block, including eight bytes for the block type and length fields, plus the number of bytes in the build number.
Build	string	Build number of the Sourcefire 3D operating system that the third party operating system string in the user input is mapped to.

User Product Data Block Fields (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the patch number of the Sourcefire 3D operating system that the third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the patch String data block, including eight bytes for the block type and length fields, plus the number of bytes in the patch number.
Patch	string	Patch number of the Sourcefire 3D operating system that the third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the extension number of the Sourcefire 3D operating system that the third party operating system string is mapped. This value is always 0.
String Block Length	uint32	Number of bytes in the extension String data block, including eight bytes for the block type and length fields, plus the number of bytes in the extension number.
Extension	string	Extension number of the Sourcefire 3D operating system that the third party operating system string in the user input is mapped to.
UUID	uint8 [x16]	Contains the unique identification number for the operating system.
String Block Type	uint32	Initiates a String data block containing the device hardware information in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the build String data block, including eight bytes for the block type and length fields, plus the number of bytes in the build number.
Device String	string	Mobile device hardware information.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.

User Product Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Fix List data blocks conveying user input data regarding what fixes have been applied to hosts in the specified IP address ranges. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Fix List data blocks.
Fix List Data Blocks *	variable	Fix List data blocks containing information about fixes applied to the hosts. See Fix List Data Block on page 279 for a description of this data block.

User Data Blocks

User data blocks appear in user event messages. They are a subset of the series 1 data blocks. For information on the general format of series 1 data blocks, see Understanding Discovery (Series 1) Blocks on page 224.

IMPORTANT! The data block length field of the user data block header contains the number of bytes in the data block, including the eight bytes of the two data block header fields.

The User Data Block Type table lists the user data blocks that can appear in user event messages. Data blocks are listed by data block type. Current data blocks are the latest versions. Legacy blocks are supported but not produced by the current version of the Sourcefire 3D System.

User Data Block Type

Type	Content	Data Block Category	Description
73	User Login Information	Legacy	Contains changes in login information for users detected by the system. See User Login Information Data Block 5.1+ on page 378 for more information. The successor block type introduced for version 5.0 has the same structure as block type 73 but with different data in the fields.
74	User Account Update Message	Current	Contains changes in user account information. See User Account Update Message Data Block on page 364 for more information.
75	User Information for 4.7 - 4.10.x	Legacy	Contains changes in information for users detected by the system. See User Information Data Block on page 375 for more information. The successor block type 120 introduced for version 5.0 has the same structure as block type 75.
120	User Information for 5.0+	Current	Contains changes in information for users detected by the system. See User Information Data Block on page 375 for more information. Supersedes block type 75.
121	User Login Information	Legacy	Contains changes in login information for users detected by the system. See User Login Information Data Block for 5.0 - 5.0.2 on page 565 for more information. Differs from block 73 in the content of the Protocol field, which stores the Version 5.0+ application ID for the application protocol ID detected in the event. The successor block introduced for version 5.1 has block type 127.

User Data Block Type (Continued)

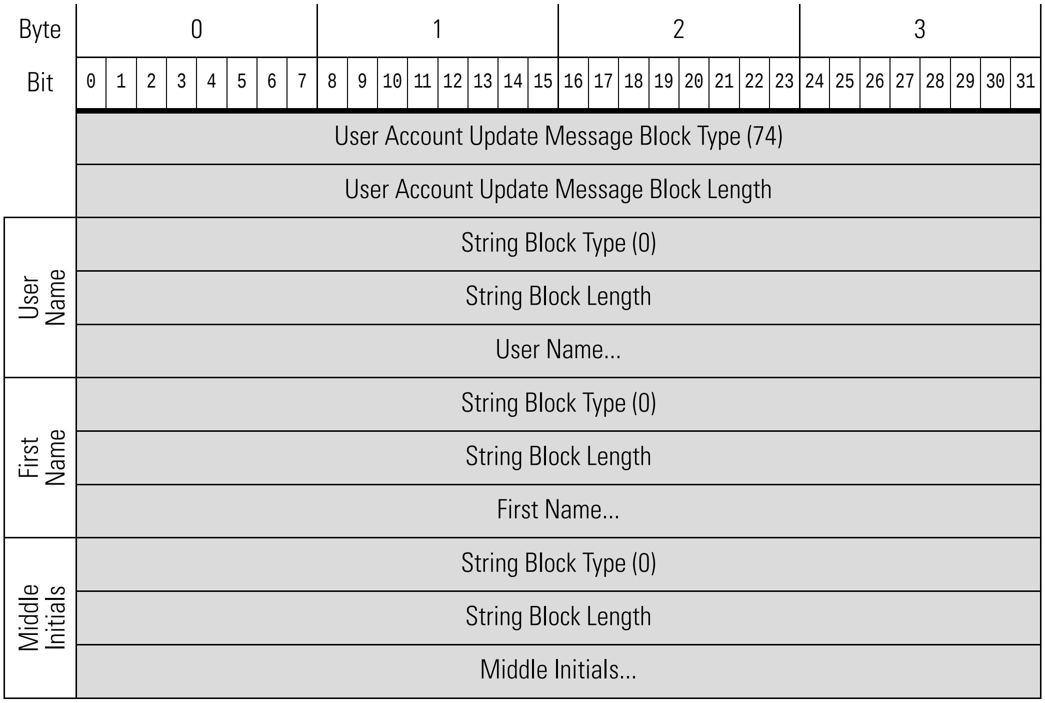
TYPE	CONTENT	DATA BLOCK CATEGORY	DESCRIPTION
127	User Login Information	Current	Contains changes in login information for users detected by the system. See User Login Information Data Block 5.1+ on page 378 for more information. It supersedes block type 121.
151	IOC State	Current	Contains information about compromises. See IOC State Data Block for 5.3+ on page 158 for more information.

User Account Update Message Data Block

The User Account Update Message data block conveys information about updates to a user's account information.

The User Account Update Message data block has a block type of 74 in the series 1 group of blocks.

The following diagram shows the format of the User Account Update Message data block:

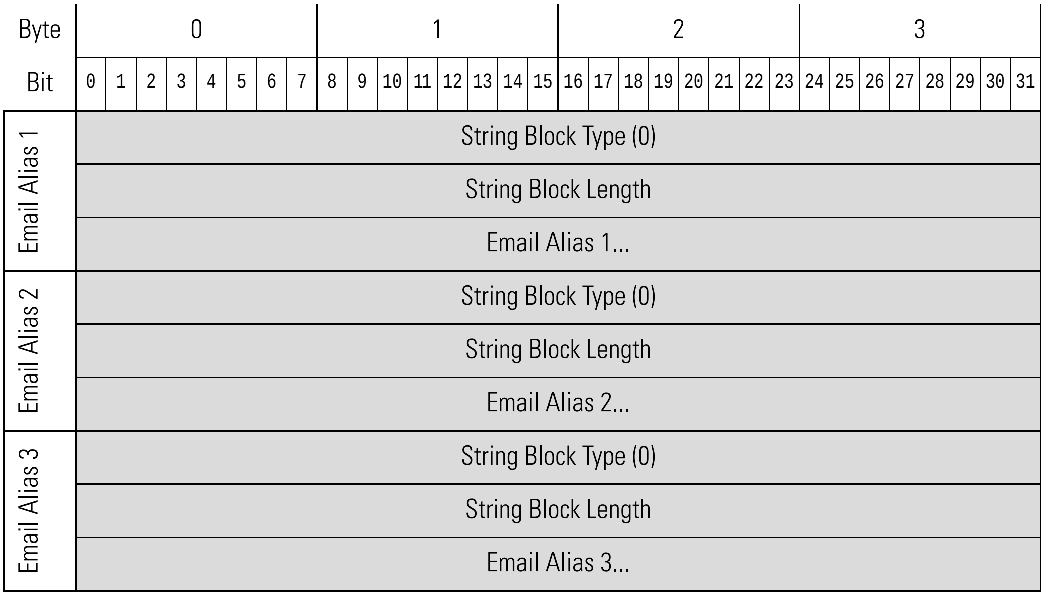


	User Account Update Message Block Type (74)
	User Account Update Message Block Length
User Name	String Block Type (0)
	String Block Length
	User Name...
First Name	String Block Type (0)
	String Block Length
	First Name...
Middle Initials	String Block Type (0)
	String Block Length
	Middle Initials...

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Last Name	String Block Type (0)			
	String Block Length			
	Last Name...			
Full Name	String Block Type (0)			
	String Block Length			
	Full Name...			
Title	String Block Type (0)			
	String Block Length			
	Title...			
Staff Identity	String Block Type (0)			
	String Block Length			
	Staff Identity...			
Address	String Block Type (0)			
	String Block Length			
	Address...			
City	String Block Type (0)			
	String Block Length			
	City...			
State	String Block Type (0)			
	String Block Length			
	State...			
Country/ Region	String Block Type (0)			
	String Block Length			
	Country/Region...			

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Postal Code	String Block Type (0)																															
	String Block Length																															
	Postal Code...																															
Building	String Block Type (0)																															
	String Block Length																															
	Building...																															
Location	String Block Type (0)																															
	String Block Length																															
	Location...																															
Room	String Block Type (0)																															
	String Block Length																															
	Room...																															
Company	String Block Type (0)																															
	String Block Length																															
	Company...																															
Division	String Block Type (0)																															
	String Block Length																															
	Division...																															
Dept	String Block Type (0)																															
	String Block Length																															
	Department...																															
Office	String Block Type (0)																															
	String Block Length																															
	Office...																															

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Mailstop		String Block Type (0)			
		String Block Length			
		Mailstop...			
Email		String Block Type (0)			
		String Block Length			
		Email...			
Phone		String Block Type (0)			
		String Block Length			
		Phone...			
IP Phone		String Block Type (0)			
		String Block Length			
		IP Phone...			
User 1		String Block Type (0)			
		String Block Length			
		User 1...			
User 2		String Block Type (0)			
		String Block Length			
		User 2...			
User 3		String Block Type (0)			
		String Block Length			
		User 3...			
User 4		String Block Type (0)			
		String Block Length			
		User 4...			



The User Account Update Message Data Block Fields table describes the components of the User Account Update Message data block.

User Account Update Message Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
User Account Update Message Block Type	uint32	Initiates a User Account Update Message data block. This value is always 74.
User Account Update Message Block Length	uint32	Total number of bytes in the User Account Update Message data block, including eight bytes for the user account update message block type and length fields, plus the number of bytes in the user account update message data that follows.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The username for the user.

User Account Update Message Data Block Fields (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the first name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the first name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the first name.
First Name	string	The first name for the user.
String Block Type	uint32	Initiates a String data block containing the middle initials for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the middle initials String data block, including eight bytes for the block type and length fields, plus the number of bytes in the middle initials.
Middle Initials	string	The middle initials for the user.
String Block Type	uint32	Initiates a String data block containing the last name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the last name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the last name.
Last Name	string	The last name for the user.
String Block Type	uint32	Initiates a String data block containing the full name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the full name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the full name.
Full Name	string	The full name for the user.
String Block Type	uint32	Initiates a String data block containing the title for the user. This value is always 0.

User Account Update Message Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the title String data block, including eight bytes for the block type and length fields, plus the number of bytes in the title.
Title	string	The title for the user.
String Block Type	uint32	Initiates a String data block containing the staff identification for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the staff identity String data block, including eight bytes for the block type and length fields, plus the number of bytes in the staff identity.
Staff Identity	string	The staff identity for the user.
String Block Type	uint32	Initiates a String data block containing the address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the address.
Address	string	The address for the user.
String Block Type	uint32	Initiates a String data block containing the city from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the city String data block, including eight bytes for the block type and length fields, plus the number of bytes in the city.
City	string	The city from the user's address.
String Block Type	uint32	Initiates a String data block containing the state from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the state String data block, including eight bytes for the block type and length fields, plus the number of bytes in the state.
State	string	The state for the user.

User Account Update Message Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block containing the country or region from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the country or region String data block, including eight bytes for the block type and length fields, plus the number of bytes in the country or region.
Country or Region	string	The country or region from the user's address.
String Block Type	uint32	Initiates a String data block containing the postal code from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the postal code String data block, including eight bytes for the block type and length fields, plus the number of bytes in the postal code.
Postal Code	string	The postal code from the user's address.
String Block Type	uint32	Initiates a String data block containing the building from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the building String data block, including eight bytes for the block type and length fields, plus the number of bytes in the building name.
Building	string	The building from the user's address.
String Block Type	uint32	Initiates a String data block containing the location from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the location String data block, including eight bytes for the block type and length fields, plus the number of bytes in the location name.
Location	string	The location from the user's address.
String Block Type	uint32	Initiates a String data block containing the room from the user's address. This value is always 0.

User Account Update Message Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the room String data block, including eight bytes for the block type and length fields, plus the number of bytes in the room.
Room	string	The room from the user's address.
String Block Type	uint32	Initiates a String data block containing the company from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the company String data block, including eight bytes for the block type and length fields, plus the number of bytes in the company name.
Company	string	The company from the user's address.
String Block Type	uint32	Initiates a String data block containing the division from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the division String data block, including eight bytes for the block type and length fields, plus the number of bytes in the division name.
Division	string	The division from the user's address.
String Block Type	uint32	Initiates a String data block containing the department from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the department String data block, including eight bytes for the block type and length fields, plus the number of bytes in the department.
Department	string	The department from the user's address.
String Block Type	uint32	Initiates a String data block containing the office from the user's address. This value is always 0.

User Account Update Message Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the office String data block, including eight bytes for the block type and length fields, plus the number of bytes in the office.
Office	string	The office from the user's address.
String Block Type	uint32	Initiates a String data block containing the mailstop from the user's address. This value is always 0.
String Block Length	uint32	Number of bytes in the mailstop String data block, including eight bytes for the block type and length fields, plus the number of bytes in the mailstop.
Mailstop	string	The mailstop from the user's address.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
String Block Type	uint32	Initiates a String data block containing the phone number for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the phone number.
Phone	string	The phone number for the user.
String Block Type	uint32	Initiates a String data block containing the internet phone number for the user. This value is always 0.

User Account Update Message Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the internet phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the internet phone number.
Internet Phone	string	The internet phone number for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 1	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 2	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 3	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an alternate user name for the user. This value is always 0.

User Account Update Message Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the user String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
User 4	string	An alternate user name for the user.
String Block Type	uint32	Initiates a String data block containing an email alias for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email alias String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email alias.
Email alias 1	string	An email alias for the user.
String Block Type	uint32	Initiates a String data block containing an email alias for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email alias String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email alias.
Email alias 2	string	An email alias for the user.
String Block Type	uint32	Initiates a String data block containing an email alias for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email alias String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email alias.
Email alias 3	string	An email alias for the user.

User Information Data Block

The User Information data block is used in User Modification messages and conveys information for a user detected, removed, or dropped. For more information, see User Modification Messages on page 223

The User Information data block has a block type of 75 in the series 1 group of blocks for version 4.7 - 4.10.x and a block type of 120 in the series 1 group of blocks for 5.0+. The structures are the same for block types 75 and 120.

The following diagram shows the format of the User Information data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	User Information Block Type (75 | 120)			
	User Information Block Length			
	User ID			
User Name	String Block Type (0)			
	String Block Length			
	User Name...			
	Protocol			
First Name	String Block Type (0)			
	String Block Length			
	First Name...			
Last Name	String Block Type (0)			
	String Block Length			
	Last Name...			
Email	String Block Type (0)			
	String Block Length			
	Email...			
Department	String Block Type (0)			
	String Block Length			
	Department...			
Phone	String Block Type (0)			
	String Block Length			
	Phone...			

The User Information Data Block Fields table describes the components of the User Information data block.

User Information Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
User Information Block Type	uint32	Initiates a User Information data block. This value is 75 for version 4.7 - 4.10.x and a value of 120 for 5.0+.
User Information Block Length	uint32	Total number of bytes in the User Information data block, including eight bytes for the user information block type and length fields plus the number of bytes in the user information data that follows.
User ID	uint32	Identification number of the user.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields plus the number of bytes in the username.
Username	string	The username for the user.
Protocol	uint32	The protocol for the packet containing the user information.
String Block Type	uint32	Initiates a String data block containing the first name of the user. This value is always 0.
String Block Length	uint32	Number of bytes in the first name String data block, including eight bytes for the block type and length fields plus the number of bytes in the first name.
First Name	string	The first name for the user.
String Block Type	uint32	Initiates a String data block containing the last name for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the user last name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the last name.

User Information Data Block Fields (Continued)

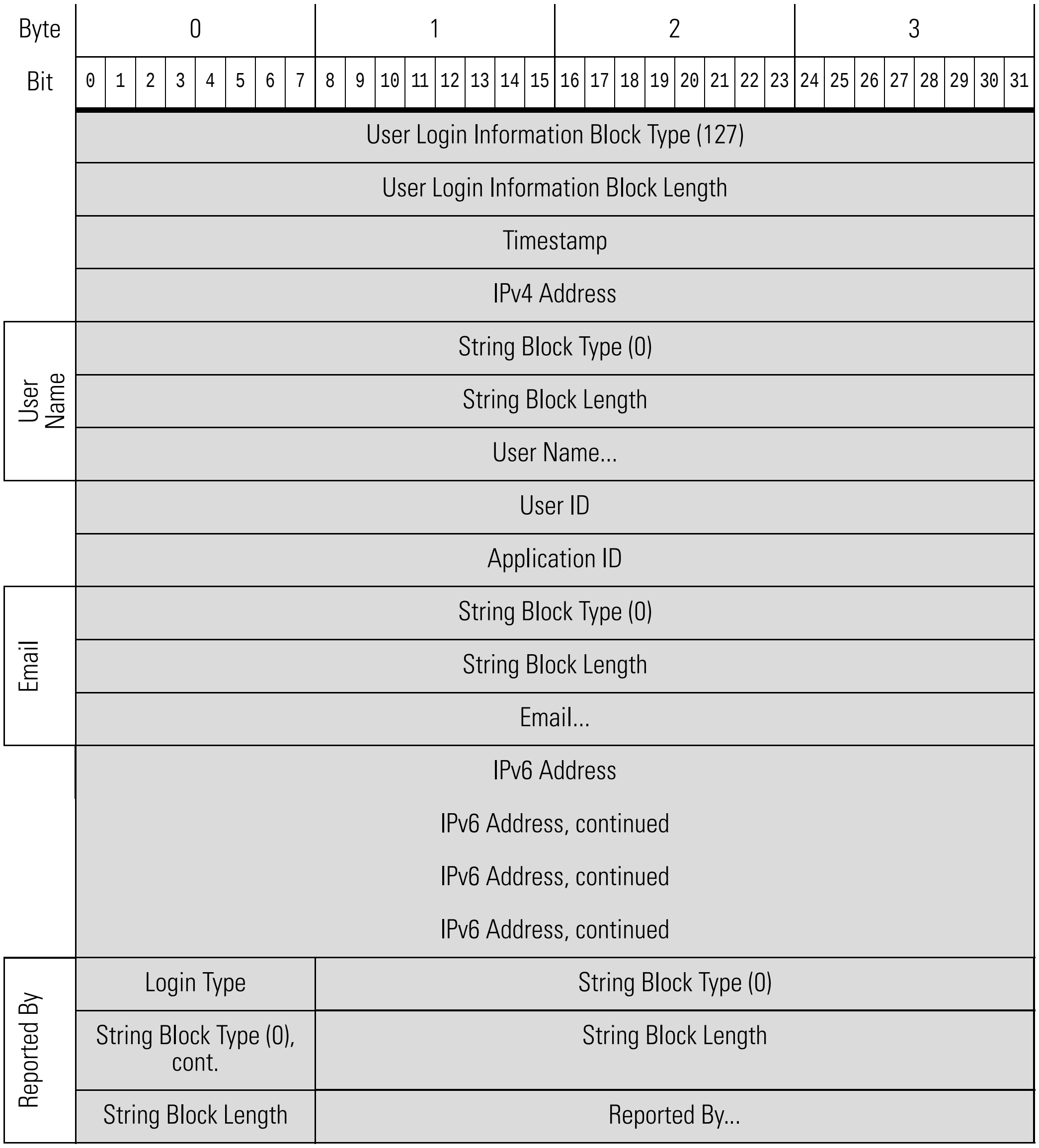
FIELD	DATA TYPE	DESCRIPTION
Last Name	string	The last name for the user.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.
String Block Type	uint32	Initiates a String data block containing the department for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the department String data block, including eight bytes for the block type and length fields, plus the number of bytes in the department.
Department	string	The department for the user.
String Block Type	uint32	Initiates a String data block containing the phone number for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the phone number String data block, including eight bytes for the block type and length fields, plus the number of bytes in the phone number.
Phone	string	The phone number for the user.

User Login Information Data Block 5.1+

The User Login Information data block is used in User Information Update messages and conveys changes in login information for a detected user. For more information, see User Information Update Message Block on page 223.

The User Login Information data block has a block type of 73 for version 4.7 - 4.10.x, a block type of 121 in the series 1 group of blocks for version 5.0 - 5.0.2, and a block type of 127 in the series 1 group of blocks for version 5.1+.

The graphic below shows the format of the User Login Information data block:



The User Login Information Data Block Fields table describes the components of the User Login Information data block.

User Login Information Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
User Login Information Block Type	uint32	Initiates a User Login Information data block. This value is 127 for version 5.1+.
User Login Information Block Length	uint32	Total number of bytes in the User Login Information data block, including eight bytes for the user login information block type and length fields, plus the number of bytes in the user login information data that follows.
Timestamp	uint32	Timestamp of the event.
IPv4 Address	uint32	IPv4 address from the host where the user was detected logging in, in IP address octets.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The user name for the user.
User ID	uint32	Identification number of the user.
Application ID	uint32	The application ID for the application protocol used in the connection that the login information was derived from.
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.

User Login Information Data Block Fields (Continued)

Field	Data Type	Description
IPv6 Address	uint8[16]	IPv6 address from the host where the user was detected logging in, in IP address octets.
Login Type	uint8	The type of user login detected.
String Block Type	uint32	Initiates a String data block containing the Reported By value. This value is always 0.
String Block Length	uint32	Number of bytes in the Reported By String data block, including eight bytes for the block type and length fields, plus the number of bytes in the Reported By field.
Reported By	string	The name of the Active Directory server reporting a login.

Discovery and Connection Event Series 2 Data Blocks

In the Discovery and Connection Event Series 2 Block Types table below, the Data Block Status field indicates whether the block is current (the latest version) or legacy (used in an older version and can still be requested through eStreamer).

Discovery and Connection Event Series 2 Block Types

Type	Content	Data Block Status	Description
15	Access Control Rule	Current	Used by access control rule metadata messages to map policy UUID and rule ID values to a descriptive string. See Access Control Rule Data Block on page 382.
21	Access Control Rule Reason	Current	Used by access control rule metadata messages to map access control rule reasons to a descriptive string. See Access Control Rule Reason Data Block 5.1+ on page 383.
22	Security Intelligence Category	Current	Used to store Security Intelligence information. See Security Intelligence Category Data Block 5.1+ on page 385.

Access Control Rule Data Block

The eStreamer service uses the Access Control Rule data block in access control rule metadata messages to map policy UUID and rule ID combinations to a descriptive string. The Access Control Rule data block has a block type of 15 in the series 2 group of blocks.

The following graphic shows the structure of the Access Control Rule data block.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Access Control Rule Block Type (15)			
	Access Control Rule Block Length			
	Access Control Rule UUID Access Control Rule UUID, continued Access Control Rule UUID, continued Access Control Rule UUID, continued			
	Access Control Rule ID			
	String Block Type (0)			
	String Block Length			
	Name...			

The Access Control Rule Data Block Fields table describes the fields in the Access Control Rule data block.

Access Control Rule Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Access Control Rule Block Type	uint32	Initiates an Access Control Rule block. This value is always 15.
Access Control Rule Block Length	uint32	Total number of bytes in the Access Control Rule block, including eight bytes for the Access Control Rule block type and length fields, plus the number of bytes of data that follows.
Access Control Rule UUID	uint8[16]	The unique identifier for the access control rule.

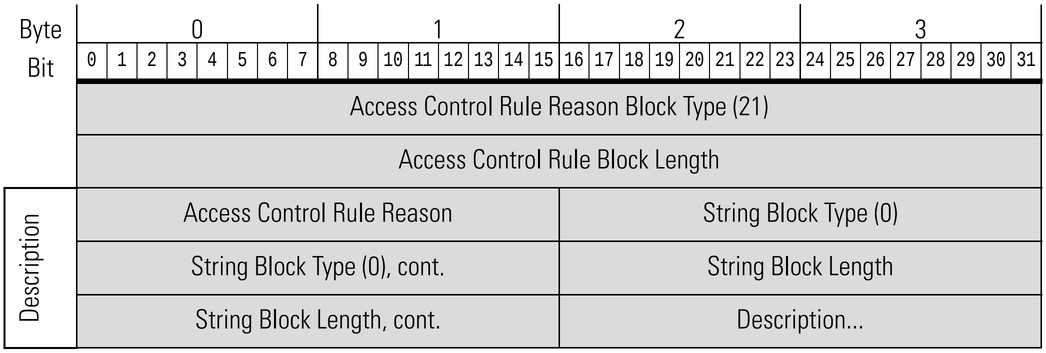
Access Control Rule Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Access Control Rule ID	uint32	The internal Sourcefire identifier for the access control rule.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule UUID and access control rule ID. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
Name	string	The descriptive name.

Access Control Rule Reason Data Block 5.1+

The eStreamer service uses the Access Control Rule Reason data block in Access Control Rule Reason metadata messages to map Access Control reasons to a descriptive string. The Access Control Rule Reason data block has a block type of 21 in the series 2 group of blocks.

The following graphic shows the structure of the Access Control Rule Reason data block.:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Access Control Rule Reason Block Type (21)			
	Access Control Rule Block Length			
Description	Access Control Rule Reason		String Block Type (0)	
	String Block Type (0), cont.		String Block Length	
	String Block Length, cont.		Description...	

The Access Control Rule Reason Data Block Fields table describes the fields in the Access Control Rule Reason data block.

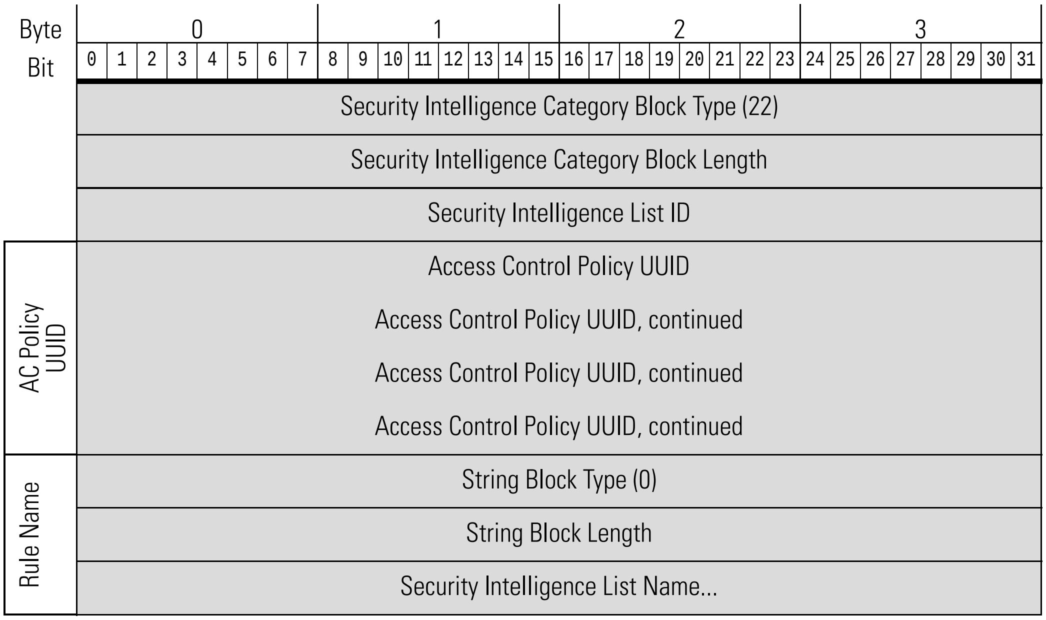
Access Control Rule Reason Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Access Control Rule Reason Block Type	uint32	Initiates an Access Control Rule Reason block. This value is always 21.
Access Control Rule Reason Block Length	uint32	Total number of bytes in the Access Control Rule Reason block, including eight bytes for the Access Control Rule Reason block type and length fields, plus the number of bytes of data that follows.
Access Control Rule Reason	uint16	The reason the Access Control rule logged the connection.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Description field.
Description	string	Description of the Access Control rule reason.

Security Intelligence Category Data Block 5.1+

The eStreamer service uses the Security Intelligence Category data block in access control rule metadata messages to stream Security Intelligence information. The Security Intelligence Category data block has a block type of 22 in the series 2 group of blocks.

The following graphic shows the structure of the Security Intelligence Category data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Security Intelligence Category Block Type (22)			
	Security Intelligence Category Block Length			
	Security Intelligence List ID			
AC Policy UUID	Access Control Policy UUID			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
Rule Name	String Block Type (0)			
	String Block Length			
	Security Intelligence List Name...			

The Security Intelligence Category Data Block fields table describes the fields in the Security Intelligence Category data block.

Security Intelligence Category Data Block fields

FIELD	DATA TYPE	DESCRIPTION
Security Intelligence Category Block Type	uint32	Initiates an Security Intelligence Category data block. This value is always 22.
Security Intelligence Category Block Length	uint32	Total number of bytes in the Security Intelligence Category block, including eight bytes for the Security Intelligence Category block type and length fields, plus the number of bytes of data that follows.

Security Intelligence Category Data Block fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Security Intelligence List ID	uint32	The ID of the IP blacklist or whitelist triggered by the connection.
Access Control Policy UUID	uint8[16]	The UUID of the access control policy configured for Security Intelligence.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the access control rule reason. This value is always 0.
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Security Intelligence List Name field.
Security Intelligence List Name	string	The name of the Security Intelligence category IP blacklist or whitelist triggered by the connection.

CHAPTER 5

UNDERSTANDING HOST DATA STRUCTURES

This chapter describes the format of the Full Host Profile data block that conveys a set of data describing a single host. The eStreamer server generates and sends these blocks on request for host data. For information about the client request procedure, the message structure, and the delivery method, see Host Data and Multiple Host Data Message Format on page 51.

eStreamer uses the series 1 data block structure to package these Full Host profile blocks. For the general structure of series 1 blocks, see Series 1 Data Block Header on page 224. The Full Host Profile data block contains a number of encapsulated blocks which are individually described in the subsections where they are defined in Understanding Discovery & Connection Data Structures on page 164.

See the following sections for more information about current and legacy Full Host Profile data blocks:

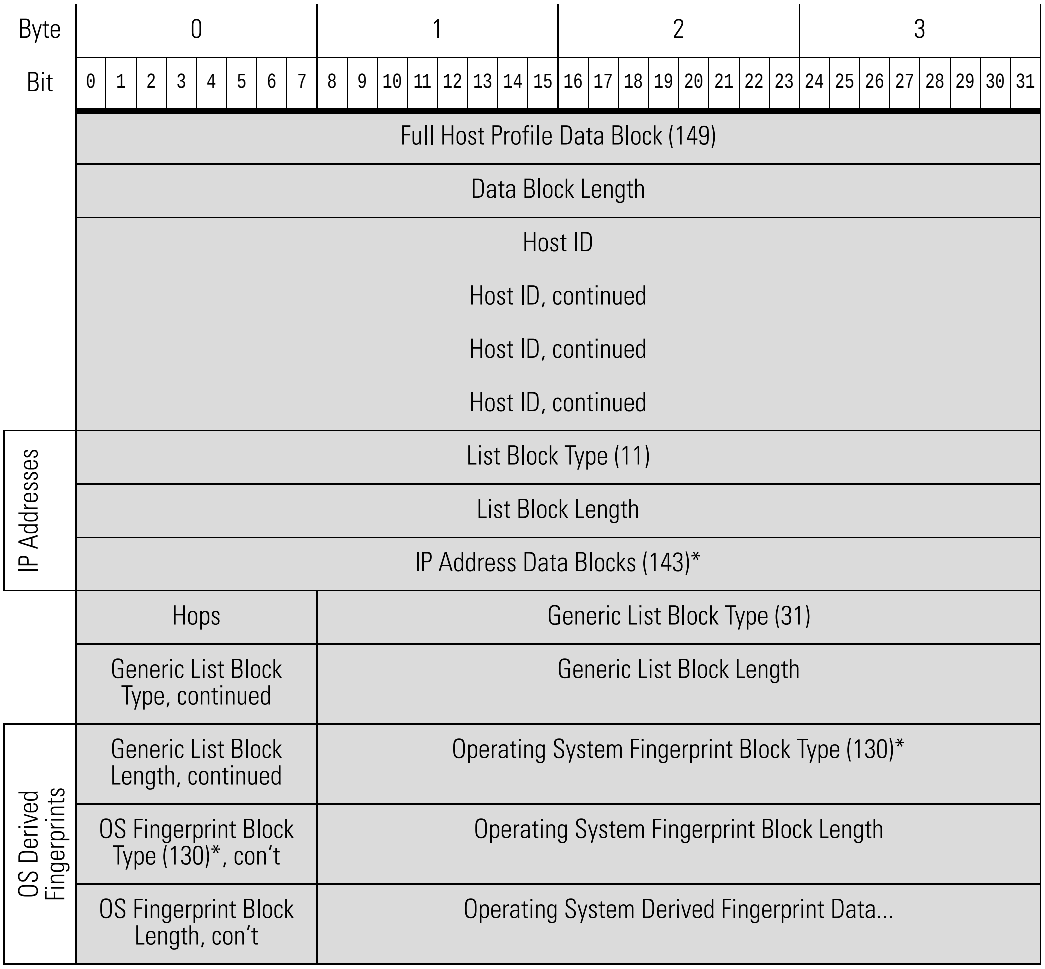
- Full Host Profile Data Block 5.3+ on page 388 describes the current Full Host Profile data block structure.
- Full Host Profile Data Block 5.0 - 5.0.2 on page 673 describes the legacy Full Host Profile data block structure for versions 5.0 - 5.0.2.
- Full Host Profile Data Block 4.8 on page 656 describes the legacy Full Host Profile data block structure for versions 4.9 - 4.10.x.

Full Host Profile Data Block 5.3+

The Full Host Profile data block for version 5.3+ contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures on page 164. The Full Host Profile data block a block type value of 149. It supersedes the prior version, which has a block type of 140.

IMPORTANT! An asterisk (*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Host Profile data block for 5.3+:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Host Profile Data Block (149)			
	Data Block Length			
	Host ID			
	Host ID, continued			
	Host ID, continued			
	Host ID, continued			
IP Addresses	List Block Type (11)			
	List Block Length			
	IP Address Data Blocks (143)*			
	Hops	Generic List Block Type (31)		
	Generic List Block Type, continued	Generic List Block Length		
OS Derived Fingerprints	Generic List Block Length, continued	Operating System Fingerprint Block Type (130)*		
	OS Fingerprint Block Type (130)*, con't	Operating System Fingerprint Block Length		
	OS Fingerprint Block Length, con't	Operating System Derived Fingerprint Data...		

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Generic List Block Type (31)			
	Generic List Block Length			
Server Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Server Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Client Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Client Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System VDB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System VDB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
User Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System User Fingerprint Data...			
	Generic List Block Type (31)			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Generic List Block Length			
Scan Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Scan Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Application Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Application Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Conflict Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Mobile Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Mobile Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
IPv6 Server Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System IPv6 Server Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Ipv6 Client Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Ipv6 Client Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Ipv6 DHCP Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System IPv6 DHCP Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
User Agent Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System User Agent Fingerprint Data...			
(TCP) Full Server Data	List Block Type (11)...			
	List Block Length...			
	(TCP) Full Server Data Blocks (104)*			
(UDP) Full Server Data	List Block Type (11)			
	List Block Length			
	(UDP) Full Server Data Blocks (104)*			
Network Protocol Data	List Block Type (11)			
	List Block Length			
	(Network) Protocol Data Blocks (4)*			
Transport Protocol Data	List Block Type (11)			
	List Block Length			
	(Transport) Protocol Data Blocks (4)*			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
MAC Address Data	List Block Type (11)			
	List Block Length			
	Host MAC Address Data Blocks (95)*			
	Last Seen			
	Host Type			
	Business Criticality		VLAN ID	
	VLAN Type	VLAN Priority	Generic List Block Type (31)	
Host Client Data	Generic List Block Type, continued		Generic List Block Length	
	Generic List Block Length, continued		Full Host Client Application Data Blocks (112)*	
NetBios Name	String Block Type (0)			
	String Block Length			
	NetBIOS Name String...			
Notes Data	String Block Type (0)			
	String Block Length			
	Notes String....			
(VDB) Host Vulns	Generic List Block Type (31)			
	Generic List Block Length			
	(VDB) Host Vulnerability Data Blocks (85)*			
3rd Pty/VDB) Host Vulns	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*			
3rd Pty Scan Host Vulns	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*			

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td rowspan="3">Attribute Value Data</td><td colspan="32">List Block Type (11)</td></tr>
<tr><td colspan="32">List Block Length</td></tr>
<tr><td colspan="32">Attribute Value Data Blocks *</td></tr>
<tr><td></td><td colspan="8">Mobile</td><td colspan="8">Jailbroken</td><td colspan="16">Generic List Block Type (31)</td></tr>
<tr><td rowspan="2">IOC State</td><td colspan="16">Generic List Block Type, continued</td><td colspan="16">Generic List Block Length</td></tr>
<tr><td colspan="16">Generic List Block Length, continued</td><td colspan="16">IOC State Data Blocks (150)*</td></tr>
</table>

The Full Host Profile Record 5.3+ Fields table describes the components of the Full Host Profile for 5.3+ record.

Full Host Profile Record 5.3+ Fields

Field	Data Type	Description
Host ID	uint8[16]	Unique ID number of the host. This is a UUID.
List Block Type	uint32	Initiates a List data block comprising IP address data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated IP address data blocks.
IP Address	variable	IP addresses of the host and when each IP address was last seen. See Host IP Address Data Block on page 273 for a description of this data block.
Hops	uint8	Number of network hops from the host to the device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying mobile device fingerprint data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Mobile) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a mobile device host. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 DHCP) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a user agent fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Agent) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a user agent fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.

Full Host Profile Record 5.3+ Fields (Continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — host • 1 — router • 2 — bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+ on page 331 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Sourcefire-detected IDs. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block on page 253 for a description of the data blocks in this list.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IOC State data blocks.This value is always 31.

Full Host Profile Record 5.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IOC State data blocks.
IOC State Data Blocks *	variable	IOC State data blocks containing information about compromises on a host. See IOC State Data Block for 5.3+ on page 158 for a description of this data block.

CHAPTER 6

CONFIGURING ESTREAMER

After you create a client application, you can connect it to the eStreamer server, start the eStreamer service, and begin exchanging data.

IMPORTANT! An *eStreamer server* is a Defense Center or Device (version 4.9 or higher) where the eStreamer service is running.

Perform the following tasks to manage eStreamer and client interaction:

1. Enable eStreamer on the eStreamer server.

   See Configuring eStreamer on the eStreamer Server on page 405 for information about allowing access to the eStreamer server, adding clients, and generating authentication credentials to establish an authenticated connection.

2. If required, manually run the eStreamer service (eStreamer). You can stop, start, and view the status of the service, and use command line options to debug client-server communication.

   See Managing the eStreamer Service on page 412 for more information.

3. Optionally, to use the eStreamer reference client to troubleshoot a connection or data stream, set up the reference client on the computer where you plan to run your client.

   See Configuring the eStreamer Reference Client on page 414.

Configuring eStreamer on the eStreamer Server

LICENSE: Any

Before the Defense Center or Device you want to use as an eStreamer server can begin streaming events to a client application, you must configure the eStreamer server to send events to clients, provide information about the client, and generate a set of authentication credentials to use when establishing communication. You can perform all of these tasks from the Defense Center or Device user interface.

See the following sections for more information:

- Configuring eStreamer Event Types on page 405
- Adding Authentication for eStreamer Clients on page 407
- Using An Alternate Management Interface with eStreamer on page 409

Configuring eStreamer Event Types

LICENSE: Any

You can control which types of events the eStreamer server is able to transmit to client applications that request them.

Available event types on a Device or a Defense Center include:

- Intrusion events
- Intrusion event packet data
- Intrusion event extra data

Available event types on a Defense Center include:

- Discovery events (this also enables connection events)
- Correlation and white list events
- Impact flag alerts
- User activity events
- Malware events
- File events

Note that the primary and secondary in a stacked 3D9900 pair report intrusion events to the Defense Center as if they were separate managed devices. If you configure communication with an eStreamer client on the primary in a 3D9900 stack, you also need to configure the client on the secondary; the client configuration is not replicated. Similarly, when you delete the client, delete it in both places. If you configure an eStreamer client for a Defense Center managing 3D9900s in a stack configuration, note that the Defense Center reports all events received from both managed devices, even if the same event is reported by both.

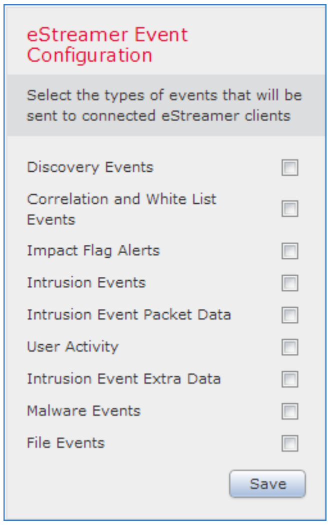
If you configure an eStreamer client on a Defense Center in a high availability configuration, the client configuration is not replicated from the primary Defense Center to the secondary Defense Center.

To configure the types of events captured by eStreamer:

ACCESS: Admin

1. Select **System > Local > Registration**.
2. Click **eStreamer**.

   The eStreamer page appears with the **eStreamer Event Configuration** menu.



eStreamer Event Configuration

Select the types of events that will be sent to connected eStreamer clients

Discovery Events □
Correlation and White List Events □
Impact Flag Alerts □
Intrusion Events □
Intrusion Event Packet Data □
User Activity □
Intrusion Event Extra Data □
Malware Events □
File Events □

3. Select the check boxes next to the types of events you want eStreamer to capture and forward to requesting clients. Note that if a check box is currently unchecked, that data is not being captured. Unchecking a check box does not delete data that has already been captured.

You can select any or all of the following on a Device or Defense Center:

- **Intrusion Events** to transmit intrusion events generated by managed devices.
- **Intrusion Event Packet Data** to transmit packets associated with intrusion events.
- **Intrusion Event Extra Data** to transmit additional data associated with intrusion events, such as the URI associated with the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

You can also select any or all of the following on a Defense Center:

- **Discovery Events** to transmit host discovery events

TIP! If you want connection events, then you must enable discovery events.

- **Correlation Events** to transmit correlation and white list events.
- **Impact Flag Alerts** to transmit impact alerts generated by the Defense Center.
- **User Activity Events** to transmit user events.
- **Intrusion Event Extra Data** to transmit additional data for intrusion events, such as the URI associated with the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

IMPORTANT! Note that this controls which events the eStreamer server can transmit. Your client application must still specifically request the types of events you want it to receive. For more information, see Request Flags on page 30.

4. Click **Save**.

Your settings are saved and the events you selected will be forwarded to eStreamer clients when requested.

Adding Authentication for eStreamer Clients

LICENSE: Any

Before eStreamer can send events to a client, you must add the client to the eStreamer server’s peers database. You must also copy the authentication certificate generated by the eStreamer server to the client.

To add an eStreamer client:

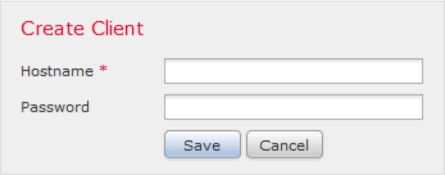
ACCESS: Admin

1. Select **Local** > **Registration** > **eStreamer**.

   The **eStreamer** page appears.

2. Click **Create Client**.

   The Create Client page appears.

   Create Client

   Hostname *

   Password

3. In the **Hostname** field, enter the host name or IP address of the host running the eStreamer client.

   IMPORTANT! If you use a host name, the host input server **must** be able to resolve the host to an IP address. If you have not configured DNS resolution, you should configure it first or use an IP address.

4. If you want to encrypt the certificate file, enter a password in the **Password** field.

5. Click **Save**.

   The eStreamer server allows the client computer to access port 8302 on the Defense Center and creates an authentication certificate to use during client-server authentication. The eStreamer Client page re-appears, with the new client listed under **eStreamer Clients**.

6. Click the download icon (⬇) next to the certificate file.

7. Save the certificate file to the directory used by your client computer for SSL authentication.

   The client can now connect to the Defense Center.

   TIP! To revoke access for a client, click the delete icon (🗑) next to the host you want to remove. Note that you do not need to restart the host input service on the Defense Center; access is revoked immediately.

Using An Alternate Management Interface with eStreamer

LICENSE: Any

By default, eStreamer uses the primary management interface, `eth0` to stream data to eStreamer clients. You can configure eStreamer to use the any available management interface. This example uses `eth1`, but you can use any available management interface. eStreamer can only use one interface at a time.

To configure eStreamer to use an alternate management interface

ACCESS: Admin (escalated to Root)

1. Connect the `eth1` interface on the Defense Center to a broadcast domain that is the same domain as the eStreamer clients and not the same domain as the `eth0` interface.

   WARNING! Placing the `eth0` and `eth1` interfaces in the same broadcast domain will impact performance.

2. Open an SSH connection to the Defense Center.
3. Log into the Defense Center.
4. Type `sudo su` to get root access. If necessary, type your password again when prompted.
5. Type the command `configure-network -i eth1 enable` to enable the `eth1` interface.
6. Configure the `eth1` interface using the prompts.

   The following prompt appears:
   `Do you wish to configure IPv4? (y or n)`
7. Type `y` and press Enter. If you do not type `y`, you will not be prompted to configure IPv4 or IPv6.

   The following prompt appears:
   `Management IP address? [10.5.60.199]`
8. Type the desired IP address and press Enter.

   The following prompt appears:
   `Management netmask? [255.255.0.0]`
9. Type the netmask and press Enter.

   The following prompt appears:
   `Management default gateway? [10.5.1.1]`
10. The default gateway is loaded from the `eth0` interface. You cannot set the default gateway specifically for eStreamer.

11. Press Enter to accept the default gateway from the `eth0` interface.

The following prompt appears:

```
Are these settings correct? (y or n)
```

12. Check the settings. If they are correct, type `y` and press Enter.

The following prompt appears:

```
Do you wish to configure IPv6? (y or n)
```

13. IPv6 is not supported for this feature. Type `n` and press Enter.

The Defense Center confirms that the `eth1` interface is configured:

```
Updated network configuration.
```

14. After you configure `eth1`, type the command `manage_estreamer.pl` to start the eStreamer configuration utility.

The following menu appears:

```
****************  Configuration Utility  **************

1   status
2   disable
3   enable
4   restart
5   send IDS events
6   don't send IDS events
7   send Packet Data
8   don't send Packet Data
9   add Client
10  delete Client
11  set EStreamer Interface
0   Exit

************************************************************
Enter choice:

****************  Configuration Utility  **************
```

15. Type choice `11` and press Enter.

16. Type `eth1` and press Enter.

```
************************************************************
Enter choice: 11
Enter Interface Name% eth1
Reloading EStreamer... at
/usr/local/sf/lib/perl/5.10.1/SF/EStreamer.pm line 449.

****************  Configuration Utility  **************
```

If you have not configured `eth1`, you will receive the warning:

```
IPv4 or IPv6 addresses are not assigned to the interface: eth1.
```

17. Type `0` and press Enter to exit the script.

    At this point, eStreamer will send information out via the `eth1` interface rather than `eth0`.

    WARNING! The script can be run to configure eStreamer to use the `eth1` interface even if the `eth1` interface is not connected. However, eStreamer cannot stream data over the interface unless the eStreamer client is on a host that is reachable from that interface.

To configure eStreamer to use the primary management interface

ACCESS: Root

1. Open an SSH connection to the Defense Center.
2. Log into the Defense Center.
3. Type `sudo su` to get root access. If necessary, type your password again when prompted.
4. Type the command `manage_estreamer.pl` to start the eStreamer configuration utility. This will bring up the following menu:

```
****************  Configuration Utility  **************

1   status
2   disable
3   enable
4   restart
5   send IDS events
6   don't send IDS events
7   send Packet Data
8   don't send Packet Data
9   add Client
10  delete Client
11  set EStreamer Interface
0   Exit

************************************************************
Enter choice:

****************  Configuration Utility  **************
```

5. Type choice `11` and press Enter.

6. Type `eth0` and press Enter.

```
************************************************************
Enter choice: 11
Enter Interface Name% eth0
Reloading EStreamer... at
/usr/local/sf/lib/perl/5.10.1/SF/EStreamer.pm line 449.
****************  Configuration Utility  **************
```

7. Type `0` and press Enter to exit the script.

8. Type the command `configure-network -i eth1 disable` to disable the `eth1` interface.

Managing the eStreamer Service

LICENSE: Any

You can manage the eStreamer service from the user interface. However, you can also use the command line to start and stop the service. The following sections describe eStreamer command line options:

- Starting and Stopping the eStreamer Service on page 412 describes how to start and stop the eStreamer service.
- eStreamer Service Options on page 413 describes the command line options available for the eStreamer service and how to use them.

Starting and Stopping the eStreamer Service

LICENSE: Any

You can manage the eStreamer service using the `manage_estreamer.pl` script, which allows you to start, stop, reload, and restart the service.

TIP! You can also add command line options to the eStreamer initialization script. See eStreamer Service Options on page 413 for more information.

The eStreamer Management Options table describes the options in the `manage_estreamer.pl` script you can use on the Defense Center or Device.

eStreamer Management Options

OPTION	DESCRIPTION	SELECT OPTION NUMBER....
enable	Starts the service.	3
disable	Stops the service.	2

eStreamer Management Options (Continued)

Option	Description	Select option Number...
restart	Restarts the service.	`4`
status	Indicates whether the service is running.	`1`

eStreamer Service Options

License: Any

eStreamer provides many service options that allow you to troubleshoot the service. You can use the options described in the eStreamer Service Options table with the eStreamer service.

eStreamer Service Options

Option	Description
`--debug`	Runs eStreamer with debug-level logging. Errors are saved in the syslog and (when used in conjunction with **`--nodaemon`**) appear on screen.
`--nodaemon`	Runs eStreamer as a foreground process. Errors appear on-screen.
`--nohostcheck`	Runs eStreamer with host name checking disabled. That is, if the client host name does not match the host name contained in the subjectAltName:dNSName entry in the client certificate, access is still allowed. The **`nohostcheck`** option is useful in cases where the network DNS and/or NAT configuration prevent the host name check from succeeding. Note that all other security checks are performed. **WARNING!** Enabling this option can negatively affect the security of your system.

Use the above options by first stopping the eStreamer service, then running it with the options you want, and finally restarting the service. For example, you can follow the instructions provided in Running the eStreamer Service in Debug Mode to debug eStreamer functionality.

Running the eStreamer Service in Debug Mode

LICENSE: Any

You can run the eStreamer service in debug mode to view each status message the service generates on your terminal screen. Use the following procedure to do debugging.

To run the eStreamer service in debug mode:

ACCESS: Admin

1. Log into the Defense Center or Device using SSH.
2. Use `manage_estreamer.pl` and select option `2` to stop the eStreamer service.
3. Use `./usr/local/sf/bin/sfestreamer --nodaemon --debug` to restart the eStreamer service in debug mode.

   Status messages for the service appear on the terminal screen.
4. When you are finished debugging, restart the service in normal mode using `manage_estreamer.pl` and selecting. option `4`.

Configuring the eStreamer Reference Client

The *reference client* provided with the eStreamer SDK is a set of sample client scripts and Perl modules included to illustrate how the eStreamer API can be used. You can run them to familiarize yourself with eStreamer output, or you can use them to debug problems with installations of your custom-built client.

For more information on setting up the reference client, see the following sections:

- Setting Up the eStreamer Perl Reference Client on page 414
- Running the eStreamer Perl Reference Client on page 422

Setting Up the eStreamer Perl Reference Client

To use the eStreamer Perl reference client, you must first configure the sample scripts to fit your environment and requirements.

For more information, see the following sections:

- Understanding the eStreamer Perl Reference Client on page 415
- Configuring Communications for the eStreamer Reference Client on page 416
- Loading General Prerequisites for the Perl Reference Client on page 416
- Downloading and Unpacking the Perl Reference Client on page 416
- Understanding the Data Requested by a Test Script on page 417

- Modifying the Type of Data Requested by a Test Script on page 418
- Creating a Certificate for the Perl Reference Client on page 420

Understanding the eStreamer Perl Reference Client

You can download the eStreamerSDK.zip package, which contains the eStreamer Perl reference client, from the Sourcefire support site. The following files are included in the eStreamerSDK.zip package:

- `SF_CUSTOM_ALERT.MIB`

  This MIB file is used by the `snmp.pm` file to set up traps for SNMP.
- SFRecords.pm

  This Perl module contains definitions of discovery message record blocks.
- SFStreamer.pm

  This Perl module contains the functions called by the Perl clients.
- SFPkcs12.pm

  This Perl module parses the client certificate and allows the client to connect to the eStreamer server.
- SFRNABlocks.pm

  This Perl module contains definitions of discovery data blocks.
- ssl_test.pl

  You can use this Perl script to test an intrusion event request over an SSL connection.
- OutputPlugins/csv.pm

  This Perl module prints intrusion events to a comma-separated value (CSV) format.
- OutputPlugins/print.pm

  This Perl module prints events to a human-readable format.
- OutputPlugins/snmp.pm

  This Perl module sends events to the specified SNMP server.
- OutputPlugins/pcap.pm

  This Perl module stores packet captures as a pcap file.
- OutputPlugins/syslog.pm

  This Perl module sends events to the local syslog server.

Configuring Communications for the eStreamer Reference Client

The reference client uses the Secure Sockets Layer (SSL) for data communication. You must install OpenSSL on the computer you plan to use as a client and configure it appropriately for your environment.

IMPORTANT! For initial installations on Linux operating systems, you must install the `libssl-dev` component as part of this download.

To set up SSL on your client:

1. Download OpenSSL from http://openssl.org/source/.
2. Unpack the source to `/usr/local/src`.
3. Configure the source by running the Configure script.
4. Make and install the compiled source.

Loading General Prerequisites for the Perl Reference Client

Before you can run the eStreamer Perl reference client, you must install the `IO::Socket::SSL` Perl module on the client computer. You can install the module manually or use `cpan` to do so.

IMPORTANT! If the `Net::SSLeay` module is not installed on the client computer, install that module as well. `Net::SSLeay` is required for communication with OpenSSL.

You also need to install and configure OpenSSL to support an SSL connection to the eStreamer server. For more information, see Configuring Communications for the eStreamer Reference Client on page 416.

Loading Prerequisites for the Perl SNMP Reference Client

Before you can run the Estreamer SNMP module of the Perl reference client, you must install the latest `net-snmp` Perl modules available for the client operating system on the client computer.

Downloading and Unpacking the Perl Reference Client

You can download the `EventStreamerSDK.zip` file that contains the eStreamer Perl reference client from https://support.sourcefire.com.

Unpack the zip file to a computer running the Linux operating system, where you plan to run the client.

Understanding the Data Requested by a Test Script

By default, when you use the ssl_test -o setting in the reference client, you request data as indicated in the following table.

Default Requests Made by Output Plugins

THIS SYNTAX...	CALLS PLUGIN...	AND SENDS...	TO REQUEST THE FOLLOWING DATA...
`./ssl_test.pl` *`eStreamerServerName`* `-h HostIPAddresses`	N/A	Host request, message type 5, with bit 11 set to 1	Host data (see Host Data and Multiple Host Data Message Format on page 51)
`./ssl_test.pl` *`eStreamerServerName`* `-o print -f` *`TextFile`*	OutputPlugins/ print.pm	Event stream request, message type 2, with bits 2 and 20-24 set to 1	Event data (see Event Stream Request Message Format on page 28, on page 106, Correlation Policy Record on page 85, Correlation Rule Record on page 87, Metadata for Discovery Events on page 172, Host Discovery Structures by Event Type on page 205, and User Data Structures by Event Type on page 222) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o pcap -f` *`TargetPCAPFile`*	OutputPlugins/ pcap.pm	Event stream request, message type 2, with bits 0 and 23 set to 1	Packet data (see Event Data Message Format on page 37 and Packet Record 4.8.0.2+ on page 67) eStreamer transmits only packet data because bit 0 is set on the event stream request.

Default Requests Made by Output Plugins (Continued)

THIS SYNTAX...	CALLS PLUGIN...	AND SENDS...	TO REQUEST THE FOLLOWING DATA...
`./ssl_test.pl` *`eStreamerServerName`* `-o csv -f` *`CSVFile`*	OutputPlugins/ csv.pm	Event stream request, message type 2, with bits 2 and 23 set to 1	Intrusion event data (see Event Data Message Format on page 37 and Intrusion Event Record 5.3+ on page 70) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o snmp -f` *`SNMPServer`*	OutputPlugins/ snmp.pm	Event stream request, message type 2, with bits 2, 20, and 23 set to 1	Intrusion event data (see Event Data Message Format on page 37 and Intrusion Event Record 5.3+ on page 70) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.
`./ssl_test.pl` *`eStreamerServerName`* `-o syslog`	OutputPlugins/ syslog.pm	Event stream request, message type 2, with bits 2, 20, and 23 set to 1	Intrusion event data (see Event Data Message Format on page 37 and Intrusion Event Record 5.3+ on page 70) eStreamer transmits type 1 intrusion events because bit 2 is set on the event stream request.

Modifying the Type of Data Requested by a Test Script

The `SFStreamer.pm` Perl module defines several request flag variables that you can use in the sample scripts to request data. The following table indicates what request flag variable to call to set each request flag in an event stream request message. If you want to request different data using one of the output modules, you can edit the $FLAG settings in the module.

For more information on the request flags, the data they request, and the product versions corresponding to each flag, see Request Flags on page 30.

Request Flag Variables Used in Sample Scripts

VARIABLE	SETS REQUEST FLAG...	TO REQUEST THE FOLLOWING DATA...
$FLAG_PKTS	0	Packet data
$FLAG_METADATA	1	Version 1 metadata
$FLAG_IDS	2	Type 1 intrusion events
$FLAG_RNA	3	Version 1 discovery events
$FLAG_POLICY_EVENTS	4	Version 1 correlation events
$FLAG_IMPACT_ALERTS	5	Intrusion impact alerts
$FLAG_IDS_IMPACT_FLAG	6	Type 7 intrusion events
$FLAG_RNA_EVENTS_2	7	Version 2 discovery events
$FLAG_RNA_FLOW	8	Version 1 connection data
$FLAG_POLICY_EVENTS_2	9	Version 2 correlation events
$FLAG_RNA_EVENTS_3	10	Version 3 discovery events
$FLAG_HOST_ONLY	11	When sent in conjunction with $FLAG_HOST_SINGLE (for one host) or $FLAG_HOST_MULTI (for multiple hosts), only host data with no event data
$FLAG_RNA_FLOW_3	12	Version 3 connection data
$FLAG_POLICY_EVENTS_3	13	Version 3 correlation events
$FLAG_METADATA_2	14	Version 2 metadata
$FLAG_METADATA_3	15	Version 3 metadata
$FLAG_RNA_EVENTS_4	17	Version 4 discovery events
$FLAG_RNA_FLOW_4	18	Version 4 connection data

Request Flag Variables Used in Sample Scripts (Continued)

VARIABLE	SETS REQUEST FLAG...	TO REQUEST THE FOLLOWING DATA...
$FLAG_POLICY_EVENTS_4	19	Version 4 correlation events
$FLAG_METADATA_4	20	Version 4 metadata
$FLAG_RUA	21	User activity events
$FLAG_POLICY_EVENTS_5	22	Version 5 correlation events
$FLAGS_SEND_ARCHIVE_ TIMESTAMP	23	Extended event headers that include the timestamp applied when the event was archived for eStreamer server to process
$FLAG_RNA_EVENTS_5	24	Version 5 discovery events
$FLAG_RNA_EVENTS_6	25	Version 6 discovery events
$FLAG_RNA_FLOW_5	26	Version 5 connection data
$FLAG_EXTRA_DATA	27	Intrusion event extra data record
$FLAG_RNA_EVENTS_7	28	Version 7 discovery events
$FLAG_POLICY_EVENTS_6	29	Version 6 correlation events
$FLAG_DETAIL_REQUEST	30	Extended request to eStreamer

WARNING! In all event types, prior to version 5.x, the reference client labels `detection engine ID` fields as `sensor ID`.

Creating a Certificate for the Perl Reference Client

LICENSE: Any

Before you can use the Perl reference client, you need to create a certificate on the Defense Center or Device for the computer where you want to run the client. You then download the certificate file to the client computer and use it to create a certificate (`server.crt`) and RSA key file (`server.key`).

To create a certificate for the Perl Reference Client:

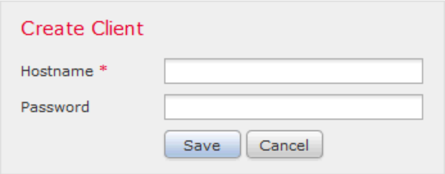
ACCESS: Admin

1. Select **Operations** > **Configuration** > **eStreamer**.

   The eStreamer page appears.

2. Click **Create Client**.

   The Create Client page appears.

   Create Client

   Hostname *

   Password

3. In the **Hostname** field, enter the host name or IP address of the host running the eStreamer client.

   IMPORTANT! If you use a host name, the host input server **must** be able to resolve the host to an IP address. If you have not configured DNS resolution, you should configure it first or use an IP address.

4. If you want to encrypt the certificate file, enter a password in the **Password** field.

5. Click **Save**.

   The eStreamer server allows the client computer to access port 8302 on the Defense Center and creates an authentication certificate to use during client-server authentication. The eStreamer Client page re-appears, with the new client listed under **eStreamer Clients**.

6. Click the download icon (⬇) next to the certificate file.

7. Save the certificate file to the directory used by your client computer for SSL authentication.

   The client can now connect to the Defense Center.

   TIP! To revoke access for a client, click the delete icon (🗑) next to the host you want to remove. Note that you do not need to restart the host input service on the Defense Center; access is revoked immediately.

Running the eStreamer Perl Reference Client

The eStreamer Perl reference client scripts are designed for use on a 64-bit operating system with the Linux kernel but should work on any POSIX-based 64-bit operating system, as long as the client machine meets the prerequisites defined in Setting Up the eStreamer Perl Reference Client on page 414.

For more information, see the following sections:

- Testing a Client Connection over SSL Using a Host Request on page 422
- Capturing a PCAP Using the Reference Client on page 422
- Capturing CSV Records Using the Reference Client on page 423
- Sending Records to an SNMP Server Using the Reference Client on page 423
- Logging Events to the Syslog Using the Reference Client on page 423
- Connecting to an IPv6 Address on page 423

Testing a Client Connection over SSL Using a Host Request

You can use the `ssl_test.pl` script to test the connection between the eStreamer server and the eStreamer client. The `ssl_test.pl` script handles any record type and prints it to STDOUT or to an output plugin you specify. When you use the `-h` option without an output option, it streams host data for the specified hosts to your terminal.

IMPORTANT! You cannot use this script to stream packet data without directing it to an output plugin because printing raw packet data to STDOUT interferes with your terminal.

Use the following syntax to use the `ssl_test.pl` script to send host data to the standard output:

```
./ssl_test.pl eStreamerServerIPAddress -h HostIPAddresses
```

For example, to test receipt of host data for the hosts in the 10.0.0.0/8 subnet over a connection to an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -h 10.0.0.0/8
```

Capturing a PCAP Using the Reference Client

You can use the reference client to capture streamed packet data in a PCAP file to see the structure of the data the client receives. Note that you must use `-f` to specify a target file when you use the `-o pcap` output option.

Use the following syntax to capture streamed packet data in a PCAP file using the `ssl_test.pl` script:

```
./ssl_test.pl eStreamerServerIPAddress -o pcap
   -f ResultingPCAPFile
```

For example, to create a PCAP file named **test.pcap** using events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o pcap -f test.pcap
```

Capturing CSV Records Using the Reference Client

You can also use the reference client to capture streamed intrusion event data in a CSV file to see the structure of the data the client receives.

Use the following syntax to run the **streamer_csv.pl** script:

`./ssl_test.pl` *`eStreamerServerIPAddress`* `-o csv -f` *`ResultingCSVFile`*

For example, to create a CSV file named **test.csv** using events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o csv -f test.csv
```

Sending Records to an SNMP Server Using the Reference Client

You can also use the reference client to stream intrusion event data to an SNMP server. Use the **-f** option to indicate the name of the SNMP trap server that should receive events. Note that this output method requires a binary named **snmptrapd** in the path and therefore only works on UNIX-like systems.

Use the following syntax to send intrusion events to an SNMP server:

```
./ssl_test.pl eStreamerServerIPAddress -o snmp
   -f SNMPServerName
```

For example, to send events to an SNMP server at 10.10.0.3 using events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o snmp -f 10.10.0.3
```

Logging Events to the Syslog Using the Reference Client

You can also use the reference client to stream intrusion events to the local syslog server on the client.

Use the following syntax to send events to the syslog:

```
./ssl_test.pl eStreamerServerIPAddress -o syslog
```

For example, to log events streamed from an eStreamer server with an IP address of 10.10.0.4:

```
./ssl_test.pl 10.10.0.4 -o syslog
```

Connecting to an IPv6 Address

You can use the reference client to connect to a Defense Center with an IPv6 address through the primary management interface. You must have the Socket6 and IO::Socket::INET6 Perl modules installed on the client machine and use the-**ipv6** option or the shortened form **-i**.

Use the following syntax to specify an IPv6 address using the ssl_test.pl script:

```
./ssl_test.pl -ipv6 eStreamerServerIPAddress
```

or

```
./ssl_test.pl -i eStreamerServerIPAddress
```

For example, to connect to a Defense Center with the IPv6 address 2001:470:e09c:20:7c1e:5248:1bf7:2ea0 use the following:

```
./ssl_test.pl -ipv6 2001:470:e09c:20:7c1e:5248:1bf7:2ea0
```

APPENDIX A
DATA STRUCTURE EXAMPLES

This appendix contains data structure examples for selected intrusion, correlation, and discovery events. Each example is displayed in binary format to clearly display how each bit is set.

See the following sections for more information:

Intrusion Event Data Structure Examples

This section contains examples of data structures that may be transmitted by eStreamer for intrusion events. The following examples are provided:

Example of an Intrusion Event for the Defense Center 5.3 +

The following diagram shows an example event record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
E	0	1	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0	1	1	0	0	1	1	1	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1	0	0	0	1
K	0	1	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1
L	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
   | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 0 0
   | 0 0 1 1 0 1 1 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1
T  | 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0|0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1
U  | 0 0 0 1 0 0 0 1|0 1 0 0 0 1 1 1|0 0 0 0 0 0 0 1|0 0 0 0 0 0 1 0
V  | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
W  | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
X  | 0 1 0 0 0 0 1 1 1 0 1 1 0 1 1 1 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1
   | 0 1 1 0 1 0 1 1 1 0 1 1 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1 0 0 1 1
   | 0 0 0 0 1 0 1 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0
   | 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 0
Y  | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Z  | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
AA | 0 1 1 1 0 1 1 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1
AB | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 1
AC | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
AD | 1 0 1 1 0 0 1 0 0 0 1 0 0 0 1 1 1 0 1 0 1 0 1 0 0 0 1 0 0 1 0 0
   | 1 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 1
   | 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1
   | 1 0 0 0 0 0 1 0 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 1
AE | 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 0
   | 0 1 1 1 1 1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 1
   | 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1
   | 0 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0
AF | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
AG | 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 1 1 1 0 1 0
```

	0	1	1	1	1	1	0	1	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1
	1	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	1	0	0	0	1	1	1	0	1	0	1	0	1	1	1	1
	0	1	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0
AH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AI	0	1	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1
AJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	0	0	1	0	1	0
AK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AL	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0																

In the preceding example, the following event information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 238 bytes long.

C. This line indicates a record type value of 400, which represents an intrusion event record.

D. This line indicates that the event record that follows is 222 bytes long.

E. This line is the timestamp when the event was saved. In this case, it was saved on Wednesday, February 5, 2014 at 19:31:40.

F. This line is reserved for future use and is populated with zeros.

G. This line indicates that the block type is 41, which is the block type for Intrusion Event records.

H. This line indicates that the data block is 222 bytes long.

I. This line indicates that the event is collected from sensor number 2.

J. This line indicates that the event identification number is 11761.

K. This line indicates that the event occurred at second 1391628571.

L. This line indicates that the event occurred at microsecond 950840.

M. This line indicates that the rule ID number is 28069.

N. This line indicates that the event was detected by generator ID number 1, the rules engine.

O. This line indicates that the rule revision number is 1.

P. This line indicates that the classification identification number is 35.

Q. This line indicates that the priority identification number is 1.

R. This line indicates that the source IP address is 10.22.8.11. Note that this field can contain either IPv4 or IPv6 addresses.

S. This line indicates that the destination IP address is 61.55.184.10. Note that this field can contain either IPv4 or IPv6 addresses.

T. The first two bytes in this line indicate that the source port number is 65268, and the second two bytes indicate that the destination port number is 53.

U. This first byte in this line indicates that UDP (17) is the protocol used in the event. The second byte is the impact flag, which indicates that the event is red (vulnerable) since the second bit is 1; that the event caused the managed event to drop the session, that the source destination host is potentially compromised, and that there is a vulnerability mapped to the client. The third byte in this line indicates that either the source or destination host is monitored by the system and is in the network map, indicating a priority 1 event (red). The last byte indicates that the event was blocked.

V. This line contains the MPLS label, if present.

W. The first two bytes in this line indicate that the VLAN ID is 2. The last two bytes are reserved and set to 0.

X. This line contains the unique ID number for the intrusion policy.

Y. This line contains the internal identification number for the user. Since there is no applicable user, it is all zeros.

Z. This line contains the internal identification number for the web application. Since there is no web application, it is all zeros.

AA. This line contains the internal identification number for the client application, which is 2000000617.

AB. This line contains the internal identification number for the application protocol, which is 617.

AC. This line contains the unique identifier for the access control rule, which is 1.

AD. This line contains the unique identifier for the access control policy.

AE. This line contains the unique identifier for the ingress interface.

AF. This line contains unique identifier for the egress interface. Since this event was blocked, there is no egress interface and the field is populated with zeros.

AG. This line contains the unique identifier for the ingress security zone.

AH. This line contains the unique identifier for the egress security zone. Since this event was blocked, there is no egress interface and the field is populated with zeros.

AI. This line contains the Unix timestamp of the connection event associated with the intrusion event.

AJ. The first two bytes in this line indicate the numerical ID of the Snort instance on the managed device that generated the connection event. The remaining two bytes indicate the value used to distinguish between connection events that happen during the same second.

AK. The first two bytes in this line indicate the code for the country of the source host. The remaining two bytes indicate the code for the country of the destination host.

AL. This line indicates the ID number of the compromise associated with this event, if any.

Example of an Intrusion Impact Alert

The following diagram shows an example intrusion impact alert record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
I	0	1	0	0	0	0	0	0	1	1	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	0	1	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
K	1	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

O	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1	0
	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0
	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	1																

In the preceding example, the following information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 58 bytes long.

C. This line indicates a record type value of 9, which represents an intrusion impact alert record.

D. This line indicates that the data that follows is 50 bytes long.

E. This line contains a value of 20, indicating that an intrusion impact alert data block follows.

F. This line indicates that the length of the impact alert block, including the impact alert block header, is 50 bytes.

G. This line indicates that the event identification number is 201256.

H. This line indicates that the event is collected from device number 2.

I. This line indicates that the event occurred at second 1087223700.

J. This line indicates that 1 (red, vulnerable) is the impact level associated with the event.

K. This line indicates that the IP address associated with the violation event is 172.16.1.22.

L. This line indicates that there is no destination IP address associated with the violation (values are set to 0).

M. This line indicates that a string block follows, containing a string block length and a text string which, in this case, contains the impact name. For more information about string blocks, see String Data Block on page 121.

N. This line indicates that the total length of the string block, including the string block indicator and length is 18 bytes. This includes 10 bytes for the impact description and 8 bytes for the string header.

O. This line indicates that the description of the impact is “Vulnerable.”

Example of a Packet Record

The following diagram shows an example packet record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	0	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	1
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0	0	1	1	0
G	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0
H	0	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	0	0	1	1	1	0	1
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	1
L	0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0

In the preceding example, the following packet information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 989 bytes long.

C. This line indicates a record type value of 2, which represents a packet record.

D. This line indicates that the packet record that follows is 981 bytes long.

E. This line indicates that the event is collected from device number 3.

F. This line indicates that the event identification number is 195430.

G. This line indicates that the event occurred at second 1057259378.

H. This line indicates that the packet was collected at second 1057259380.

I. This line indicates that the packet was collected at microsecond 254365.

J. This line indicates that the link type is 1 (Ethernet layer).

K. This line indicates that the packet data that follows is 953 bytes long.

L. This line and the following line show the actual payload data. Note that the actual data is 953 bytes and has been truncated for the sake of this example.

Example of a Classification Record for 4.6.1+

The following diagram shows an example classification record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	0	1	0
	0	1	1	0	1	1	1	1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
	0	0	1	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	1	0	1	1	1	0	1	0	0
	0	1	1	0	1	0	0	1	0	1	1	1	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1	0	0
G	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	1
	0	0	1	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0
	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	0	1	0	0	1	1	0	1	0	1	1
	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1
	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0
	0	1	1	1	0	1	1	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1
	0	1	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	0	0	1	0	0
H	1	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	0	1	0	0	0
	1	1	0	0	1	0	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	0	1
	1	0	0	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1

I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In the preceding example, the following event information appears:

A. The first two bytes of the line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 92 bytes long.

C. This line indicates a record type value of 67, which represents a classification record.

D. This line indicates that the classification record that follows is 84 bytes long.

E. This line indicates that the Classification ID is 35.

F. The first two bytes of this line indicate that the classification name that follows it is 15 bytes long. The second two bytes begin the classification name itself, which, in this case, is "trojan-activity".

G. The first byte in this line is a continuation of the classification name described in F. The next two bytes in this line indicate that the classification description that follows it is 29 bytes long. The remaining bye begin the classification description, which, in this case, is "A Network Trojan was Detected."

H. This line indicates the classification ID number that acts as a unique identifier for the classification.

I. This line indicates the classification revision ID number that acts as a unique identifier for the classification revision, which is null because there are no revisions to the classification.

Example of a Priority Record

The following example shows a sample priority record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	0	1
	0	1	1	0	0	1	1	1	0	1	1	0	1	0	0	0																

In the preceding example, the following event information appears:

A. The first two bytes in this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 16 bytes.

C. This line indicates a record type value of 4, which represents a priority record.

D. This line indicates that the priority record that follows is 8 bytes long.

E. This line indicates that the priority ID is one.

F. The first two bytes of this line indicate that there are four bytes included in the priority name. The second two bytes plus the two bytes on the following line show the priority name itself ("high").

Example of a Rule Message Record for 4.6.1+

The following example shows a sample rule record:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1
I	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1
	0	0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1

```
    0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1
    1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 0 0 0 1 1
J   0 1 1 0 1 1 0 1 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 1
    0 0 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 1 1 1 1 1
    0 0 0 1 0 0 0 1 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1
    1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 0 0 0 1 1
K   0 1 1 0 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0
    0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 1
    0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 0 1 0 0
    0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 1
    0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 1
    0 1 1 1 0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 1 0 1 1 1 0 1 0 0
    0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 0 1 0
    0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 0
    0 1 1 0 0 1 0 1 0 1 1 0 1 1 1 0 0 1 1 1 0 1 0 0 0 1 1 0 1 0 0 1
    0 1 1 0 0 0 0 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 1
    0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 0 1
    0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 1
    0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1 1
    0 1 1 1 0 1 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0
    0 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 0
    0 1 1 0 0 1 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1
    0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1
    0 0 1 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 1 0 0 1 1 0 0 0 1 1
    0 1 1 0 1 1 1 0
```

In the preceding example, the following event information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 129 bytes.

C. This line indicates a record type value of 66, which represents a rule message record.

D. This line indicates that the rule message record that follows is 121 bytes long.

E. This line indicates that the generator identification number is 1, the rules engine.

F. This line indicates that the rule identification number is 28069.

G. This line indicates that the rule revision number is 1.

H. This line indicates that the rule identification number rendered to the Sourcefire 3D System is 28069.

I. The first two bytes of this line indicate that there are 71 bytes included in the rule text name. The second two bytes begin the unique identifier number for the rule.

J. The first two bytes of this line finish the unique identifier number of the rule. The next two bytes begin the unique identifier number for the revision of the rule.

K. The first two bytes of this line finish the unique identifier number for the revision of the rule. The second two bytes begin the text of the rule message itself. The full text of the transmitted rule message is: "APP-DETECT DNS request for potential malware SafeGuard to domain 360.cn".

Example of a Version 4.0 Correlation Policy Violation Event

The following diagram shows an example correlation policy violation record in Defense Center 4.0 format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
G	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
H	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	1	0	1	0	1	0	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	0	1	0
	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
	0	0	1	1	1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0
O	0	1	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	1
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
W	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
X	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AA	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AC	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1
AD	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
AE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
AG	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AI	0	0	0	0	1	0	0	0																								

In the preceding example, the following information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 137 bytes long.

C. This line indicates a record type value of 36, which represents a correlation policy violation record for Sourcefire 3D System 4.0.

D. This line indicates that the data that follows is 129 bytes long.

E. This line contains a value of 33, indicating that a correlation policy violation data block follows.

F. This line indicates that the length of the policy violation block, including the policy violation block header, is 129 bytes.

G. The first byte line indicates that the detection engine ID is 0, indicating that the correlation event was generated on the Defense Center. The last three bytes of this line and the first byte of the next line contains the policy event timestamp, 1,098,911,301, which is Wed, 27 Oct 2004 21:08:21 GMT.

H. The last three bytes of this line and first byte of the next line indicate that the policy event ID number is 10.

I. The last three bytes of this line and first byte of the next line indicate a policy ID of 4, which, in this case, maps to a custom correlation policy on the Defense Center.

J. The last three bytes of this line and first byte of the next line indicate rule ID of 29, which, in this case, maps to a custom correlation policy rule on the Defense Center.

K. The last three bytes of this line and first byte of the next line indicate a policy priority of 1.

L. The last three bytes of this line and first byte of the next line contain a value of 0, which indicates the beginning of a string block for the policy violation event description.

M. The last three bytes of this line and first byte of the next line indicate the length of the description. In this example, the length is 21 bytes, including the string block header and the 13 bytes in the event description. In an actual event, the length is typically much longer.

N. The first byte of this line is a continuation of the string block length, followed by 13 bytes that contain the event description. The event description has been truncated for the sake of this example. In this example, the description is "`[1:2008:4] MI`." In the actual policy violation event that this example is based on, however, the description is much longer: "`[1:2008:4] MISC CVS invalid user authentication response [Impact: Potentially Vulnerable] From sensor "is.sourcefire.com" at Thu Oct 28 17:07:19 2004 UTC [Classification: Misc Attack] [Priority: 2] {tcp} 10.1.1.24:2401-> 10.1.1.25:34174`."

O. The third byte in this line has a value of one, which indicates that the type of event that caused the correlation policy violation is an intrusion event. The fourth byte in this line indicates the identification number of the device that generated the intrusion event, in this case, this is sensor 1.

P. This line indicates that the signature ID for the rule triggered in the event is 2008.

Q. This line indicates that the generator ID for the rule that triggered in the event is 1, the intrusion Detection Engine.

R. This line indicates that the intrusion event timestamp is 1,098,911,243, which means it was generated at Wed, 27 Oct 2004 21:07:23 GMT.

S. This line indicates the microsecond the intrusion event was generated, 179,035.

T. This line indicates that the ID assigned to the intrusion event is 17,828.

U. This line indicates which of the fields that follow it are valid. Based on how the bits are set, impact flags, IP protocol, source IP, source port, destination IP, and destination port fields will have values.

V. This line indicates the impact value assigned to the event. Based on how the bits are set, the impact is Orange—Potentially Vulnerable.

W. The first byte in this line indicates that the IP protocol is 6 (TCP). The second two bytes show the network protocol, which is null. The last byte of this line and first three bytes of the next line begins the source IP string, which is 10.1.1.24.

X. The first three bytes in this line finish the source IP started in line W and the last byte shows the host type, which is null.

Y. The first two bytes in this line indicate the VLAN ID, which is null. The second two bytes begin a four-byte fingerprint ID, which is also null.

Z. The first two bytes in this line complete the fingerprint ID, the second two bytes contain the source host criticality, which is null.

AA. The first two bytes of this line indicate the source port, 2401. The second two bytes begin the string block for the source host server, which has a value of 0.

AB. The first two bytes end the string block header and the second two bytes begin the string block length. The value of the string block length is 8, indicating that only the header appears and no server description string follows.

AC. The first two bytes complete the string block length. The second two bytes begin the destination IP address, which is 10.1.1.25.

AD. The first two bytes in this line complete the destination IP address. The third byte indicates the destination host type, which is null. The fourth byte begins the two byte destination VLAN ID, which is also null.

AE. The first byte in this line completes the VLAN ID, and the second three bytes begin the four-byte destination fingerprint ID, which is null.

AF. The first byte completes the destination fingerprint ID, the second two bytes contain the destination host criticality (which is null), and the last byte begins the two byte destination port (34174).

AG. The first byte completes the destination port, and the last three bytes begin a four byte string block, which has a value of 0.

AH. The first byte contains the last byte of the string header, and the last three bytes begin a four byte string length. The value here is 8, because no destination server is included in the event.

Example of a Version 4.5 - 4.6.1 Correlation Event

The following diagram shows an example correlation event record in Defense Center 4.5 - 4.6.1format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	1	0	1

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
O	0	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0
	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	0
	0	0	1	1	0	1	0	0	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
X	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
Y	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AB	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AC	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1
AD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AF	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
AG	0	0	0	0	0	0	0	0																								

In the preceding example, the following information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 153 bytes long.

C. This line indicates a record type value of 65, which represents a correlation event record for Sourcefire 3D System 4.5.

D. This line indicates that the data that follows is 145 bytes long.

E. This line contains a value of 52, indicating that a correlation event data block follows.

F. This line indicates that the length of the correlation event block, including the correlation event block header, is 145 bytes.

G. This line indicates that the detection engine ID is 0, indicating that the correlation event was generated on the Defense Center.

H. This line contains the event timestamp, 1,098,911,301, which is Wed, 27 Oct 2004 21:08:21 GMT.

I. This line indicates that the event ID number is 10.

J. This line indicates a policy ID of 4, which, in this case, maps to a custom correlation policy on the Defense Center.

K. This line indicates a rule ID of 29, which, in this case, maps to a custom correlation policy rule on the Defense Center.

L. This line indicates a policy priority of 1.

M. This line contains a value of 0, which indicates the beginning of a string block for the policy violation event description.

N. This line indicates the length of the description. In this example, the length is 19 bytes, including the string block header and the 11 bytes in the event description. In an actual event, the length is typically much longer.

O. These three lines contain the 11-byte event description, followed by the event type. The event description has been truncated for the sake of this example. In this example, the description is “`[1:2008:4]` .” In the actual policy violation event that this example is based on, however, the description is much longer: “`[1:2008:4] MISC CVS invalid user authentication response [Impact: Potentially Vulnerable] From sensor "is.sourcefire.com" at Thu Oct 28 17:07:19 2004 UTC [Classification: Misc Attack] [Priority: 2] {tcp} 10.1.1.24:2401-> 10.1.1.25:34174`.” The fourth byte in the third line has a value of one, which indicates that the type of event that caused the policy violation is an intrusion event.

P. This line indicates the identification number of the device that generated the intrusion event, in this case, this is sensor 1.

Q. This line indicates that the signature ID for the rule triggered in the event is 2008.

R. This line indicates that the generator ID for the rule that triggered in the event is 1, the intrusion detection engine.

S. This line indicates that the intrusion event timestamp is 1,098,911,243, which means it was generated at Wed, 27 Oct 2004 21:07:23 GMT.

T. This line indicates the microsecond the intrusion event was generated, 179,035.

U. This line indicates that the ID assigned to the intrusion event is 17,828.

V. This line indicates which of the fields that follow it are valid. Based on how the bits are set, impact flags, IP protocol, source IP, source port, destination IP, and destination port fields will have values.

W. This line indicates the impact value assigned to the event. Based on how the bits are set, the impact is Orange—Potentially Vulnerable.

X. The first byte in this line indicates that the IP protocol is 6 (TCP). The second two bytes show the network protocol, which is null. The last byte of this line and first three bytes of the next line begins the source IP string, which is 10.1.1.24.

Y. The first three bytes in this line finish the source IP started in line W and the last byte shows the host type, which is null.

Z. The first two bytes in this line indicate the VLAN ID, which is null. The second two bytes and the next three lines contain the first 14 bytes of a 16-byte fingerprint UUID, which is also null.

AA. The first two bytes in this line complete the fingerprint UUID, the second two bytes contain the source host criticality, which is null.

AB. The first two bytes of this line indicate the source port, 2401. The second two bytes indicate the server ID for the source host server, which has a value of 0.

AC. This line contains the destination IP address, which is 10.1.1.25.

AD. The first byte in this line indicates the destination host type, which is null. The second and third bytes indicate the two byte destination VLAN ID, which is also null. The fourth byte and the next three lines contain the first 13 bytes of a 16-byte fingerprint UUID, which is also null.

AE. The first three bytes in this line complete the 16-byte destination fingerprint ID, which is null. The fourth byte begins the destination host criticality (which is null).

AF. The first byte in this line completes the destination host criticality (which is null). The next two bytes contain the two byte destination port (34174). The last byte begins the destination server ID, which is null.

AG. The first byte in this line completes the destination server ID, which is null.

Example of a Version 4.10 Correlation Event

The following diagram shows an example correlation event record in Defense Center 4.10 format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	1	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
O	0	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	0
	0	0	1	1	0	1	0	0	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
S	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1
W	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
AB	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
AC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
AD	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0
AE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
AH	1	0	0	0	0	1	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								

In the preceding example, the following information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 153 bytes long.

C. This line indicates a record type value of 112, which represents a correlation event record for Sourcefire 3D System 4.10.

D. This line indicates that the data that follows is 145 bytes long.

Note that bit 23 was not set in the request, so Timestamp data is not included in the example.

E. This line contains a value of 107, indicating that a correlation event data block follows.

F. This line indicates that the length of the correlation event block, including the correlation event block header, is 145 bytes.

G. This line indicates that the detection engine ID is 0, indicating that the correlation event was generated on the Defense Center.

H. This line contains the correlation event timestamp, 1,098,911,301, which is Wed, 27 Oct 2004 21:08:21 GMT.

I. This line indicates that the correlation event ID number is 10.

J. This line indicates a policy ID of 4, which, in this case, maps to a custom correlation policy on the Defense Center.

K. This line indicates a rule ID of 29, which, in this case, maps to a custom correlation policy rule on the Defense Center.

L. This line indicates a policy priority of 1.

M. This line contains a value of 0, which indicates the beginning of a string block for the event description.

N. This line indicates the length of the description. In this example, the length is 19 bytes, including the string block header and the 11 bytes in the event description. In an actual event, the length is typically much longer.

O. These three lines contain the 11-byte event description, followed by the event type. The event description has been truncated for the sake of this example. In this example, the description is “ `[1:2008:4]` .” In the actual policy violation event that this example is based on, however, the description is much longer:

"[1:2008:4] MISC CVS invalid user authentication response [Impact: Potentially Vulnerable] From sensor "is.sourcefire.com" at Thu Oct 28 17:07:19 2004 UTC [Classification: Misc Attack] [Priority: 2] {tcp} 10.1.1.24:2401-> 10.1.1.25:34174." The fourth byte in the third line has a value of one, which indicates that the type of event that caused the policy violation is an intrusion event.

P. This line indicates the identification number of the detection engine that generated the intrusion event, in this case, this is detection engine 1.

Q. This line indicates that the signature ID for the rule triggered in the event is 2008.

R. This line indicates that the generator ID for the rule that triggered in the event is 1, the intrusion detection engine.

S. This line indicates that the intrusion event timestamp is 1,098,911,243, which means it was generated at Wed, 27 Oct 2004 21:07:23 GMT.

T. This line indicates the microsecond the intrusion event was generated, 179,035.

U. This line indicates that the ID assigned to the intrusion event is 17,828.

V. This line indicates which of the fields that follow it are valid. Based on how the bits are set, impact flags, IP protocol, source IP, source port, destination IP, and destination port fields will have values.

W. The first byte in this line indicates the impact value assigned to the event. Based on how the bits are set, the impact is Orange—Potentially Vulnerable. The second byte in this line indicates that the IP protocol is 6 (TCP). The last two bytes show the network protocol, which is null.

X. The line indicates the source IP string, which is 10.1.1.24.

Y. The first byte in this line indicates the host type, which is null. The second and third bytes in this line indicate the VLAN ID, which is null. The last byte and the next three lines contain the first 13 bytes of a 16-byte fingerprint UUID, which is also null.

Z. The first three bytes in this line complete the fingerprint UUID. The last byte begins the source host criticality, which is null.

AA. The first byte of this line completes the source host criticality. The last three bytes begin the source user ID, which has a value of 9.

AB. The first byte of this line completes the source user ID. The second and third bytes indicate the source port, 2401. The last byte begins the server ID for the source host server, which has a value of 0.

AC. This line completes the server ID. The last byte in this line begins the destination IP address, which is 10.1.1.25.

AD. The first three bytes in this line complete the destination IP address. The last byte indicates the destination host type, which is null.

AE. The first two bytes in this line indicate the two byte destination VLAN ID, which is also null. The third and fourth byte and the next three lines contain the first 14 bytes of a 16-byte fingerprint UUID, which is also null.

AF. The first two bytes in this line complete the 16-byte destination fingerprint ID, which is null. The third and fourth byte indicates the destination host criticality (which is null).

AG. This line indicates the destination user ID, which has a value of 20.

AH. The first two bytes of this line contain the two byte destination port (34174). The last two bytes contain the destination server ID, which is null.

AI. The first two bytes in this line indicate the destination server ID, which is null. The third byte indicates whether the packet was blocked.

Example of a Version 5.1+ User Event

The following diagram shows an example user event record in Defense Center 5.1+ format:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
E	0	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1
L	1	1	1	1	0	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
M	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	1	1	1	0	0	0	1

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P	1	1	1	0	1	1	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0
S	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1
V	1	1	1	1	0	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0
	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0
	0	0	1	0	1	1	1	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1
	0	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	1	1	1
Z	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AA	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
AB	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1
AC	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								

In the preceding example, the following information appears:

A. The first two bytes of this line indicate the standard header value of 1. The second two bytes indicate that the message is a data message (message type four).

B. This line indicates that the message that follows is 153 bytes long.

C. This line indicates a record type value of 95, which represents a user information update message block.

D. This line indicates that the data that follows is 137 bytes long.

E. This line contains the archive timestamp. It is included since bit 23 was set. The timestamp is a Unix timestamp, stored as seconds since 1/1/1970. This time stamp is 1,391,789,354, which is Mon Feb 3 19:43:49 2014.

F. This line contains zeros and is reserved for future use.

G. This line indicates that the length of the correlation event block, including the correlation event block header, is 145 bytes.

H. This line indicates that the detection engine ID is 0, indicating that the correlation event was generated on the Defense Center.

I. This line contains the correlation event timestamp, 1,098,911,301, which is Wed, 27 Oct 2004 21:08:21 GMT.

J. This line indicates that the correlation event ID number is 10.

K. This line indicates a policy ID of 4, which, in this case, maps to a custom correlation policy on the Defense Center.

L. This line indicates a rule ID of 29, which, in this case, maps to a custom correlation policy rule on the Defense Center.

M. This line indicates a policy priority of 1.

N. This line contains a value of 0, which indicates the beginning of a string block for the event description.

O. This line indicates the length of the description. In this example, the length is 19 bytes, including the string block header and the 11 bytes in the event description. In an actual event, the length is typically much longer.

P. These three lines contain the 11-byte event description, followed by the event type. The event description has been truncated for the sake of this example. In this example, the description is " `[1:2008:4]` ." In the actual policy violation event that this example is based on, however, the description is much longer: "`[1:2008:4] MISC CVS invalid user authentication response [Impact: Potentially Vulnerable] From sensor "is.sourcefire.com" at Thu Oct 28 17:07:19 2004 UTC [Classification: Misc Attack] [Priority: 2] {tcp} 10.1.1.24:2401-> 10.1.1.25:34174`." The fourth byte in the third line has a value of one, which indicates that the type of event that caused the policy violation is an intrusion event.

Q. This line indicates the identification number of the detection engine that generated the intrusion event, in this case, this is detection engine 1.

R. This line indicates that the signature ID for the rule triggered in the event is 2008.

S. This line indicates that the generator ID for the rule that triggered in the event is 1, the intrusion detection engine.

T. This line indicates that the intrusion event timestamp is 1,098,911,243, which means it was generated at Wed, 27 Oct 2004 21:07:23 GMT.

U. This line indicates the microsecond the intrusion event was generated, 179,035.

V. This line indicates that the ID assigned to the intrusion event is 17,828.

W. This line indicates which of the fields that follow it are valid. Based on how the bits are set, impact flags, IP protocol, source IP, source port, destination IP, and destination port fields will have values.

X. The first byte in this line indicates the impact value assigned to the event. Based on how the bits are set, the impact is Orange—Potentially Vulnerable. The second byte in this line indicates that the IP protocol is 6 (TCP). The last two bytes show the network protocol, which is null.

Y. The line indicates the source IP string, which is 10.1.1.24.

Z. The first byte in this line indicates the host type, which is null. The second and third bytes in this line indicate the VLAN ID, which is null. The last byte and the next three lines contain the first 13 bytes of a 16-byte fingerprint UUID, which is also null.

AA. The first three bytes in this line complete the fingerprint UUID. The last byte begins the source host criticality, which is null.

AB. The first byte of this line completes the source host criticality. The last three bytes begin the source user ID, which has a value of 9.

AC. The first byte of this line completes the source user ID. The second and third bytes indicate the source port, 2401. The last byte begins the server ID for the source host server, which has a value of 0.

AD. This line completes the server ID. The last byte in this line begins the destination IP address, which is 10.1.1.25.

AE. The first three bytes in this line complete the destination IP address. The last byte indicates the destination host type, which is null.

AF. The first two bytes in this line indicate the two byte destination VLAN ID, which is also null. The third and fourth byte and the next three lines contain the first 14 bytes of a 16-byte fingerprint UUID, which is also null.

AG. The first two bytes in this line complete the 16-byte destination fingerprint ID, which is null. The third and fourth byte indicates the destination host criticality (which is null).

Discovery Data Structure Examples

This section contains examples of data structures that may be transmitted by eStreamer for discovery events. The following examples are provided:

- Example of a New Network Protocol Message on page 453
- Example of a New TCP Server Message on page 454

Example of a New Network Protocol Message

The following diagram illustrates a sample new network protocol message for 3.0+:

Byte	0								1								2								3									
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Header Version 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Event Msg (4)	Standard Message Header
Message Length (49B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1		
New NW Protocol Msg (13)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1		
Msg Length 41B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0		
Detection Engine ID (2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		
IP (192.168.1.10)	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0		
MAC Address (none)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reserved Bytes (0)	
Unix Sec (1047242787)	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1		
Unix MSec (973208)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	0		
Reserved Bytes (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	Event Type 1000—New	
EventSub 4-New Trans Prot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		
File Number	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	1		
File Position	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
Protocol (6—TCP)	0	0	0	0	0	1	1	0																										

Example of a New TCP Server Message

The following diagram illustrates a sample new TCP server message for 3.0:

Byte	0								1								2								3									
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Header Version 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Event Msg (4)	**Standard Message Header**
Message Length (256B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
New TCP Svc Msg (11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1		
Msg Length (248B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0		
Detection Engine ID (2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		
IP (192.168.1.10)	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0		
MAC Address (none)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reserved Bytes (0)	
Unix Sec (1047242787)	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1		
Unix MSec (973208)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	0		
Rese rved Bytes (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	Event Type 1000—New	
Event Subtype 2 -New Host	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		
File Number	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	1		
File Position	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0		
Server Block Header (12)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	**Start Server Data Block**	
Server Length (208B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0		
Server Port (80)	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Hits	
Hits (1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length	
String Block Length (13B)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	1	1	1	0	1	0	0		
Server Name (`https`)	0	1	1	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length	
String Block Length (15B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	1		
Server Vendor (`Apache` + null byte)	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	1	0	1	1	0	1	0	0	0		
	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header	

Byte	0								1								2								3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Length (8-no product)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Block Length (22B)	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0	
Version - 1.3.26 (Unix)	0	0	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	
	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	1	1	1	0	
	0	1	1	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	
List Block Header (11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	**Start Sub-server List**
List Block Size (94B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	
Sub-server Hdr (1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**Start Sub-server Block**
Sub-server Len (46B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
String Length (16B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
Sub-server Name - mod_ssl	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	0	1	0	0	0	1	0	1	1	1	1	1	
	0	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
String Block Len (8B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	(No subtype vendor)
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
String Block Length (14B)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	
Sub-server Version - 2.8.9 + null character	0	0	1	1	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1	0	1	1	1	0	**End Sub-server Block**
	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**Start Sub-server Block**
Sub-server Hdr (1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sub-server Length
Sub-server Length (48B)	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Size
String Block Size (16B)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	
Sub-server Name - OpenSSL	0	1	1	0	0	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	1	
	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Header (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Data Length

Byte	0								1								2								3								
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
String Length (8-no vendor)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Header
String Block Hdr (0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	String Block Length
String Block Len (16B)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	
Sub-server Version - 0.9.6.d + null byte	0	0	1	1	1	0	0	1	0	0	1	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	1	0	1	1	1	0	**End Sub-server Block**
	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Confidence %
Confidence % (100)	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	Last used
Last Used (1047242787)	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Blob Data Block
Blob Data Block (10)	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Blob Data Length
Blob Data Length (22B)	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	1	0	0	
Server Banner (HTTP/1.1 414 Reque) -Server banner shortened for example, typically 256B.	0	1	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0	1	
	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	
	0	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	
	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	0	0	1	0	1	**End Server Data Block**

APPENDIX B
UNDERSTANDING LEGACY DATA STRUCTURES

This appendix contains information about data structures supported by eStreamer at previous versions of Sourcefire 3D System products.

If your client uses event stream requests with bits set to request data in older version formats, you can use the information in this appendix to identify the data structures of the data messages you receive.

Note that prior to version 5.0, separate detection engines were assigned IDs. For version 5.0+, devices are assigned IDs. Based on the version, data structures reflect this.

IMPORTANT! This appendix describes only data structures from version 4.9 or later of the Sourcefire 3D System. If you require documentation for structures from earlier data structure versions, contact Sourcefire Customer Support.

See the following sections for more information:

- Legacy Intrusion Data Structures on page 458
- Legacy Malware Event Data Structures on page 492
- Legacy Discovery Data Structures on page 513
- Legacy Connection Data Structures on page 577
- Legacy File Event Data Structures on page 619
- Legacy Correlation Event Data Structures on page 630
- Legacy Host Data Structures on page 656

Legacy Intrusion Data Structures

Intrusion Event (IPv4) Record for 4.9 - 4.10.x

The fields in the intrusion event (IPv4) record are shaded in the following graphic. The record type is 104 for version 4.9+, where VLAN IDs are included. The table following the graphic includes details on the fields.

You request intrusion event records by setting the intrusion event flag—bit 6 in the Request Flags field—in the request message. If you enable bit 23, an extended event header is included in the record.

Events are uniquely identified by event ID, detection device ID, and event second.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (104)			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
	Detection Engine ID			
	Event ID			
	Event Second			
	Event Microsecond			
	Rule ID (Signature ID)			
	Generator ID			
	Rule Revision			
	Classification ID			

Priority ID			
Source IPv4 Address			
Destination IPv4 Address			
Source Port/ICMP Type		Destination Port/ICMP Code	
IP Protocol ID	Impact Flags	Impact	Blocked
Reserved			
VLAN ID		Pad	

The Intrusion Event (IPv4) Record 4.9 - 4.10.x Fields table describes each intrusion event record data field.

Intrusion Event (IPv4) Record 4.9 - 4.10.x Fields

Field	Data Type	Description
Detection Engine ID	unit32	Contains the detection engine identification number.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event’s detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event’s detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IPv4 Address	uint8[4]	Source IPv4 address used in the event, in address octets.

Intrusion Event (IPv4) Record 4.9 - 4.10.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Destination IPv4 Address	uint8[4]	Destination IPv4 address used in the event, in address octets.
Source Port/ ICMP Type	uint16	If the event protocol type is TCP or UDP, this indicates the source port number. If the protocol type is ICMP, this indicates the ICMP type.
Destination Port/ICMP Code	uint16	If the event protocol type is TCP or UDP, this indicates the destination port number. If the protocol type is ICMP, this indicates the ICMP code.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event (IPv4) Record 4.9 - 4.10.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Impact Flags	bits[8]	Impact flag value of the event. The low-order seven bits are used to indicate the impact level. Values are: • 0x01 — Source or destination host is in a network monitored by the system (bit 0). • 0x02 — Source or destination host exists in the network map (bit 1). • 0x04 — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol (bit 2). • 0x08 — There is a vulnerability mapped to the operating system of the source or destination host in the event (bit 3). • 0x10 — There is a vulnerability mapped to the server detected in the event (bit 4). • 0x20 — The event caused the sensor to drop the session (used only when the sensor is running in inline mode) (bit 5). Corresponds to blocked status in Inline Result column in the Sourcefire 3D System web interface. • 0x40 — The rule that generated this event contains rule metadata setting the impact flag to red (bit 6). The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. The following impact level values map to specific priorities on the Defense Center. An **X** indicates the value can be 0 or 1: • gray (0, unknown): **0X00000** • red (1, vulnerable): **XXX1XXX, XX1XXXX, 1XXXXXX** • orange (2, potentially vulnerable): **0X00111** • yellow (3, currently not vulnerable): **0X00011** • blue (4, unknown target): **0X00001**
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — Gray (unknown impact)

Intrusion Event (IPv4) Record 4.9 - 4.10.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Blocked	uint8	Value indicating whether the event was blocked. • 0 — not blocked • 1 — blocked • 2 — would be blocked (but not permitted by configuration)
Reserved	uint32	Reserved. The display value is `MPLS Label:0`.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated. (Applies to 4.9+ events only.)
Pad	uint16	Reserved for future use.

Intrusion Event (IPv6) Record for 4.10.2.3

The fields in the intrusion event (IPv6) record are shaded in the following graphic. The record type is 105 for version 4.10.2.3, where VLAN IDs are included. The table following the graphic includes details on the fields.

You request intrusion event records by setting the intrusion event flag—bit 6 in the Request Flags field—in the request message. If you enable bit 23, an extended event header is included in the record.

Events are uniquely identified by event ID, detection device ID, and event second.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (105)			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
	Detection Engine ID			
	Event ID			
	Event Second			
	Event Microsecond			

<table>
<tr><td colspan="4">Rule ID (Signature ID)</td></tr>
<tr><td colspan="4">Generator ID</td></tr>
<tr><td colspan="4">Rule Revision</td></tr>
<tr><td colspan="4">Classification ID</td></tr>
<tr><td colspan="4">Priority ID</td></tr>
<tr><td colspan="4">Source IPv6 Address
Source IPv6 Address, continued
Source IPv6 Address, continued
Source IPv6 Address, continued</td></tr>
<tr><td colspan="4">Destination IPv6 Address
Destination IPv6 Address, continued
Destination IPv6 Address, continued
Destination IPv6 Address, continued</td></tr>
<tr><td colspan="2">Source Port/ICMP Type</td><td colspan="2">Destination Port/ICMP Code</td></tr>
<tr><td>IP Protocol ID</td><td>Impact Flags</td><td>Impact</td><td>Blocked</td></tr>
<tr><td colspan="4">Reserved</td></tr>
<tr><td colspan="2">VLAN ID</td><td colspan="2">Pad</td></tr>
</table>

The Intrusion Event (IPv6) Record 4.10.2.3+ Fields table describes each intrusion event record data field.

Intrusion Event (IPv6) Record 4.10.2.3+ Fields

FIELD	DATA TYPE	DESCRIPTION
Detection Engine ID	unit32	Contains the detection engine identification number.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.

Intrusion Event (IPv6) Record 4.10.2.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IPv6 Address	uint16[8]	Source IPv6 address used in the event, in address octets.
Destination IPv6 Address	uint16[8]	Destination IPv6 address used in the event, in address octets.
Source Port/ ICMP Type	uint16	If the event protocol type is TCP or UDP, this indicates the source port number. If the protocol type is ICMP, this indicates the ICMP type.
Destination Port/ICMP Code	uint16	If the event protocol type is TCP or UDP, this indicates the destination port number. If the protocol type is ICMP, this indicates the ICMP code.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event (IPv6) Record 4.10.2.3+ Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Impact Flags	bits[8]	Impact flag value of the event. The low-order seven bits are used to indicate the impact level. Values are: • 0x01 — Source or destination host is in a network monitored by the system (bit 0). • 0x02 — Source or destination host exists in the network map (bit 1). • 0x04 — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol (bit 2). • 0x08 — There is a vulnerability mapped to the operating system of the source or destination host in the event (bit 3). • 0x10 — There is a vulnerability mapped to the server detected in the event (bit 4). • 0x20 — The event caused the sensor to drop the session (used only when the sensor is running in inline mode) (bit 5). Corresponds to blocked status in Inline Result column in the Sourcefire 3D System web interface. • 0x40 — The rule that generated this event contains rule metadata setting the impact flag to red (bit 6). If the rule is provided by the Sourcefire Vulnerability Research Team (VRT), the source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `0X00000` • red (1, vulnerable): `XXX1XXX, XX1XXXX, 1XXXXXX` • orange (2, potentially vulnerable): `0X00111` • yellow (3, currently not vulnerable): `0X00011` • blue (4, unknown target): `0X00001`
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — Gray (unknown impact)

Intrusion Event (IPv6) Record 4.10.2.3+ Fields (Continued)

Field	Data Type	Description
Blocked	uint8	Value indicating whether the event was blocked. • 0 — not blocked • 1 — blocked • 2 — would be blocked (but not permitted by configuration)
Reserved	uint32	Reserved. The display value is `MPLS Label:0`.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated. (Applies to 4.9+ events only.)
Pad	uint16	Reserved for future use.

Intrusion Event (IPv4) Record 5.0.x - 5.1

The fields in the intrusion event (IPv4) record are shaded in the following graphic. The record type is 207.

You request intrusion event records by setting the intrusion event flag or the extended requests flag in the request message. See Request Flags on page 30 and Submitting Extended Requests on page 20.

For version 5.0.x - 5.1 intrusion events, the event ID, the managed device ID, and the event second form a unique identifier.

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Header Version (1)																Message Type (4)															
	Message Length																															
	Record Type (207)																															
	Record Length																															
	eStreamer Server Timestamp (in events, only if bit 23 is set)																															
	Reserved for Future Use (in events, only if bit 23 is set)																															
	Device ID																															
	Event ID																															
	Event Second																															
	Event Microsecond																															

Rule ID (Signature ID)			
Generator ID			
Rule Revision			
Classification ID			
Priority ID			
Source IPv4 Address			
Destination IPv4 Address			
Source Port		Destination Port	
IP Protocol ID	Impact Flags	Impact	Blocked
MPLS Label			
VLAN ID		Pad	
Policy UUID			
Policy UUID, continued			
Policy UUID, continued			
Policy UUID, continued			
User ID			
Web Application ID			
Client Application ID			
Application Protocol ID			
Access Control Rule ID			
Access Control Policy UUID			
Access Control Policy UUID, continued			
Access Control Policy UUID, continued			
Access Control Policy UUID, continued			
Interface Ingress UUID			
Interface Ingress UUID, continued			
Interface Ingress UUID, continued			
Interface Ingress UUID, continued			

Interface Egress UUID
Interface Egress UUID, continued
Interface Egress UUID, continued
Interface Egress UUID, continued
Security Zone Ingress UUID
Security Zone Ingress UUID, continued
Security Zone Ingress UUID, continued
Security Zone Ingress UUID, continued
Security Zone Egress UUID
Security Zone Egress UUID, continued
Security Zone Egress UUID, continued
Security Zone Egress UUID, continued

The Intrusion Event (IPv4) Record Fields table describes each intrusion event record data field.

Intrusion Event (IPv4) Record Fields

FIELD	DATA TYPE	DESCRIPTION
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.

Intrusion Event (IPv4) Record Fields (Continued)

Field	Data Type	Description
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IPv4 Address	uint8[4]	Source IPv4 address used in the event, in address octets.
Destination IPv4 Address	uint8[4]	Destination IPv4 address used in the event, in address octets.
Source Port	uint16	The source port number if the event protocol type is TCP or UDP.
Destination Port	uint16	The destination port number if the event protocol type is TCP or UDP.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event (IPv4) Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — Gray (unknown impact)

Intrusion Event (IPv4) Record Fields (Continued)

Field	Data Type	Description
Blocked	uint8	Value indicating whether the event was blocked. • 0 — not blocked • 1 — blocked • 2 — would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.

Intrusion Event (IPv4) Record Fields (Continued)

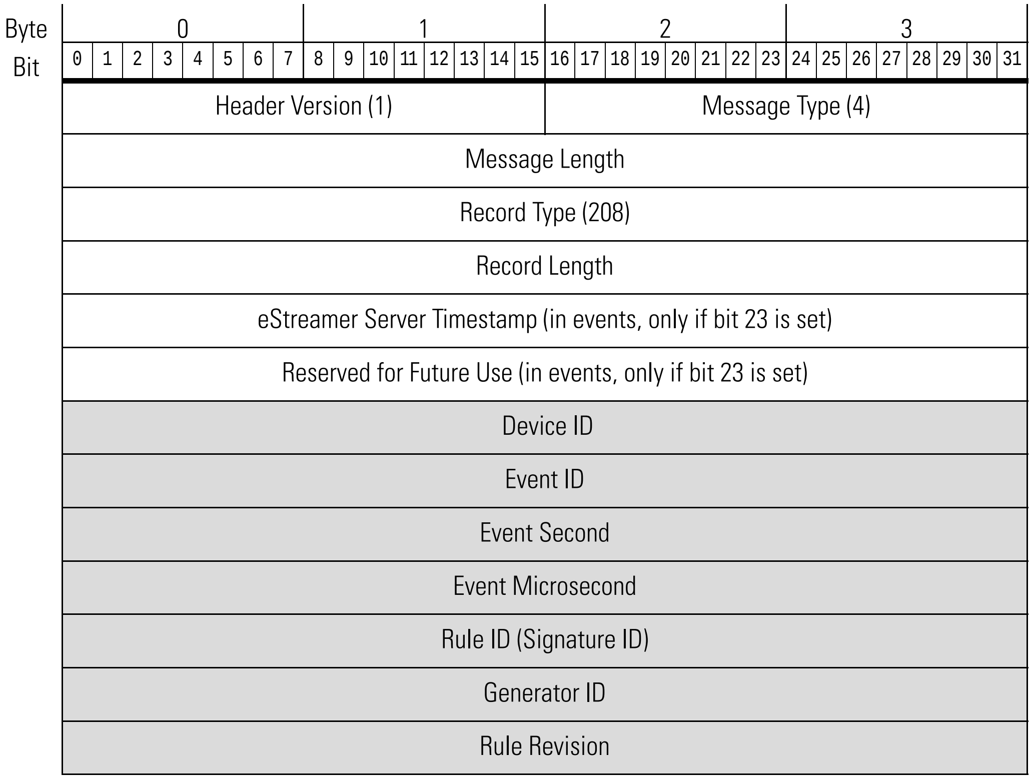
FIELD	DATA TYPE	DESCRIPTION
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.

Intrusion Event (IPv6) Record 5.0.x - 5.1

The fields in the intrusion event (IPv6) record are shaded in the following graphic. The record type is 208.

You request intrusion event records by setting the intrusion event flag or the extended requests flag in the request message. See Request Flags on page 30 and Submitting Extended Requests on page 20.

For version 5.0.x - 5.1 intrusion events, the event ID, the managed device ID, and the event second form a unique identifier.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (208)			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
	Device ID			
	Event ID			
	Event Second			
	Event Microsecond			
	Rule ID (Signature ID)			
	Generator ID			
	Rule Revision			

Classification ID			
Priority ID			
Source IPv6 Address			
Source IPv6 Address, continued			
Source IPv6 Address, continued			
Source IPv6 Address, continued			
Destination IPv6 Address			
Destination IPv6 Address, continued			
Destination IPv6 Address, continued			
Destination IPv6 Address, continued			
Source Port/ICMP Type		Destination Port/ICMP Code	
IP Protocol ID	Impact Flags	Impact	Blocked
MPLS Label			
VLAN ID		Pad	
Policy UUID			
Policy UUID, continued			
Policy UUID, continued			
Policy UUID, continued			
User ID			
Web Application ID			
Client Application ID			
Application Protocol ID			
Access Control Rule ID			
Access Control Policy UUID			
Access Control Policy UUID, continued			
Access Control Policy UUID, continued			
Access Control Policy UUID, continued			
Interface Ingress UUID			

Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued
Interface Egress UUID Interface Egress UUID, continued Interface Egress UUID, continued Interface Egress UUID, continued
Security Zone Ingress UUID Security Zone Ingress UUID, continued Security Zone Ingress UUID, continued Security Zone Ingress UUID, continued
Security Zone Egress UUID Security Zone Egress UUID, continued Security Zone Egress UUID, continued Security Zone Egress UUID, continued

The Intrusion Event (IPv6) Record Fields table describes each intrusion event record data field.

Intrusion Event (IPv6) Record Fields

Field	Data Type	Description
Device ID	unit32	Contains the identification number of the detecting device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event’s detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event’s detection.

Intrusion Event (IPv6) Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IPv6 Address	uint8[16]	Source IPv6 address used in the event, in address octets.
Destination IPv6 Address	uint8[16]	Destination IPv6 address used in the event, in address octets.
Source Port/ ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP. If the protocol type is ICMP, this indicates the ICMP type.
Destination Port/ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP. If the protocol type is ICMP, this indicates the ICMP code.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event (IPv6) Record Fields (Continued)

Field	Data Type	Description
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • 1 — Red (vulnerable) • 2 — Orange (potentially vulnerable) • 3 — Yellow (currently not vulnerable) • 4 — Blue (unknown target) • 5 — Gray (unknown impact)

Intrusion Event (IPv6) Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Blocked	uint8	Value indicating whether the event was blocked. • 0 — not blocked • 1 — blocked • 2 — would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label. (Applies to 4.9+ events only.)
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated. (Applies to 4.9+ events only.)
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.

Intrusion Event (IPv6) Record Fields (Continued)

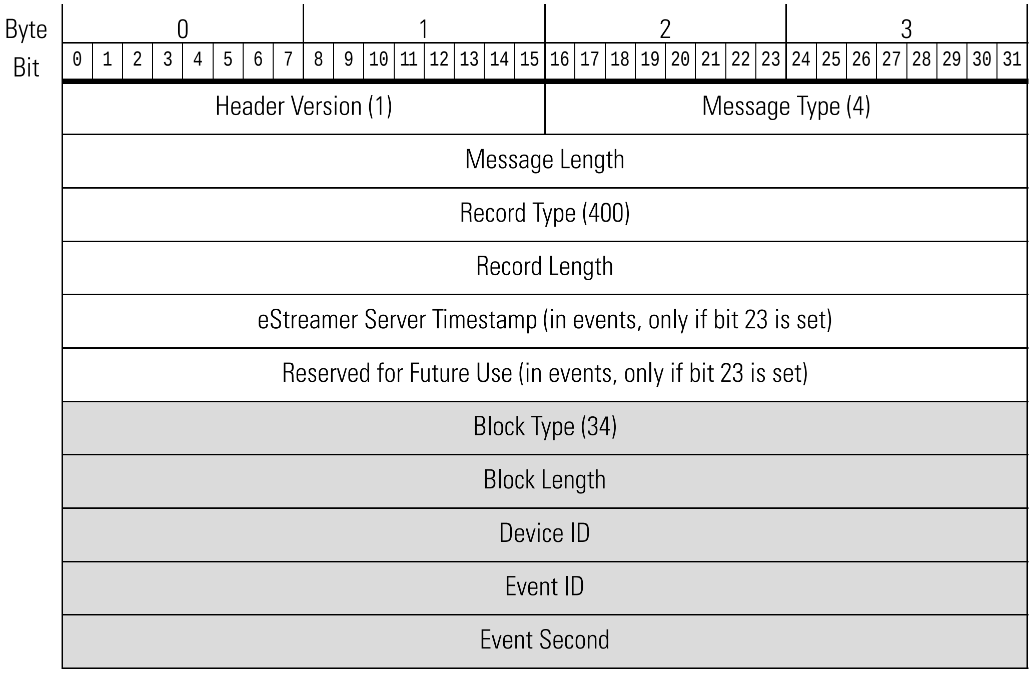
FIELD	DATA TYPE	DESCRIPTION
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.

Intrusion Event Record 5.2.x

The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 34.

You can request 5.2.x intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 5 in the Stream Request message (see Submitting Extended Requests on page 20 for information about submitting extended requests).

For version 5.2.x intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Header Version (1)		Message Type (4)	
	Message Length			
	Record Type (400)			
	Record Length			
	eStreamer Server Timestamp (in events, only if bit 23 is set)			
	Reserved for Future Use (in events, only if bit 23 is set)			
	Block Type (34)			
	Block Length			
	Device ID			
	Event ID			
	Event Second			

<table>
<tr><td colspan="4">Event Microsecond</td></tr>
<tr><td colspan="4">Rule ID (Signature ID)</td></tr>
<tr><td colspan="4">Generator ID</td></tr>
<tr><td colspan="4">Rule Revision</td></tr>
<tr><td colspan="4">Classification ID</td></tr>
<tr><td colspan="4">Priority ID</td></tr>
<tr><td colspan="4">Source IP Address</td></tr>
<tr><td colspan="4">Source IP Address, continued</td></tr>
<tr><td colspan="4">Source IP Address, continued</td></tr>
<tr><td colspan="4">Source IP Address, continued</td></tr>
<tr><td colspan="4">Destination IP Address</td></tr>
<tr><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td colspan="2">Source Port or ICMP Type</td><td colspan="2">Destination Port or ICMP Code</td></tr>
<tr><td>IP Protocol ID</td><td>Impact Flags</td><td>Impact</td><td>Blocked</td></tr>
<tr><td colspan="4">MPLS Label</td></tr>
<tr><td colspan="2">VLAN ID</td><td colspan="2">Pad</td></tr>
<tr><td colspan="4">Policy UUID</td></tr>
<tr><td colspan="4">Policy UUID, continued</td></tr>
<tr><td colspan="4">Policy UUID, continued</td></tr>
<tr><td colspan="4">Policy UUID, continued</td></tr>
<tr><td colspan="4">User ID</td></tr>
<tr><td colspan="4">Web Application ID</td></tr>
<tr><td colspan="4">Client Application ID</td></tr>
<tr><td colspan="4">Application Protocol ID</td></tr>
<tr><td colspan="4">Access Control Rule ID</td></tr>
<tr><td colspan="4">Access Control Policy UUID</td></tr>
</table>

Access Control Policy UUID, continued	
Access Control Policy UUID, continued	
Access Control Policy UUID, continued	
Interface Ingress UUID	
Interface Ingress UUID, continued	
Interface Ingress UUID, continued	
Interface Ingress UUID, continued	
Interface Egress UUID	
Interface Egress UUID, continued	
Interface Egress UUID, continued	
Interface Egress UUID, continued	
Security Zone Ingress UUID	
Security Zone Ingress UUID, continued	
Security Zone Ingress UUID, continued	
Security Zone Ingress UUID, continued	
Security Zone Egress UUID	
Security Zone Egress UUID, continued	
Security Zone Egress UUID, continued	
Security Zone Egress UUID, continued	
Connection Timestamp	
Connection Instance ID	Connection Counter
Source Country	Destination Country

The Malware Event Data Block for 5.2.x Fields table describes each intrusion event record data field.

Intrusion Event Record 5.2.x Fields

FIELD	DATA TYPE	DESCRIPTION
Block Type	unint32	Initiates an Intrusion Event data block. This value is always 34.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port or ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.

Intrusion Event Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Destination Port or ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. (version 5.0+ only) The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` (version 5.0+ only) • orange (2, potentially vulnerable): `00X0011X` • yellow (3, currently not vulnerable): `00X0001X` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — Gray (unknown impact)

Intrusion Event Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Blocked	uint8	Value indicating whether the event was blocked. • `0` — not blocked • `1` — blocked • `2` — would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.

Intrusion Event Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.

Intrusion Event Record 5.1.1.x

The fields in the intrusion event record are shaded in the following graphic. The record type is 400 and the block type is 25.

You can request 5.1.1.x intrusion events from eStreamer only by extended request, for which you request event type code 12 and version code 4 in the Stream Request message (see Submitting Extended Requests on page 20 for information about submitting extended requests).

For version 5.1.1.x intrusion events, the event ID, the managed device ID, and the event second form a unique identifier. The connection second, connection instance, and connection counter together form a unique identifier for the connection event associated with the intrusion event.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
</table>

<table>
<tr><td colspan="4">Record Type (400)</td></tr>
<tr><td colspan="4">Record Length</td></tr>
<tr><td colspan="4">eStreamer Server Timestamp (in events, only if bit 23 is set)</td></tr>
<tr><td colspan="4">Reserved for Future Use (in events, only if bit 23 is set)</td></tr>
<tr><td colspan="4">Block Type (25)</td></tr>
<tr><td colspan="4">Block Length</td></tr>
<tr><td colspan="4">Device ID</td></tr>
<tr><td colspan="4">Event ID</td></tr>
<tr><td colspan="4">Event Second</td></tr>
<tr><td colspan="4">Event Microsecond</td></tr>
<tr><td colspan="4">Rule ID (Signature ID)</td></tr>
<tr><td colspan="4">Generator ID</td></tr>
<tr><td colspan="4">Rule Revision</td></tr>
<tr><td colspan="4">Classification ID</td></tr>
<tr><td colspan="4">Priority ID</td></tr>
<tr><td colspan="4">Source IP Address
Source IP Address, continued
Source IP Address, continued
Source IP Address, continued</td></tr>
<tr><td colspan="4">Destination IP Address
Destination IP Address, continued
Destination IP Address, continued
Destination IP Address, continued</td></tr>
<tr><td colspan="2">Source Port/ICMP Type</td><td colspan="2">Destination Port/ICMP Code</td></tr>
<tr><td>IP Protocol ID</td><td>Impact Flags</td><td>Impact</td><td>Blocked</td></tr>
<tr><td colspan="4">MPLS Label</td></tr>
<tr><td colspan="2">VLAN ID</td><td colspan="2">Pad</td></tr>
<tr><td colspan="4">Policy UUID</td></tr>
</table>

Policy UUID, continued Policy UUID, continued Policy UUID, continued
User ID
Web Application ID
Client Application ID
Application Protocol ID
Access Control Rule ID
Access Control Policy UUID Access Control Policy UUID, continued Access Control Policy UUID, continued Access Control Policy UUID, continued
Interface Ingress UUID Interface Ingress UUID, continued Interface Ingress UUID, continued Interface Ingress UUID, continued
Interface Egress UUID Interface Egress UUID, continued Interface Egress UUID, continued Interface Egress UUID, continued
Security Zone Ingress UUID Security Zone Ingress UUID, continued Security Zone Ingress UUID, continued Security Zone Ingress UUID, continued
Security Zone Egress UUID Security Zone Egress UUID, continued

Security Zone Egress UUID, continued	
Security Zone Egress UUID, continued	
Connection Timestamp	
Connection Instance ID	Connection Counter

The Intrusion Event Record 5.1.1 Fields table describes each intrusion event record data field.

Intrusion Event Record 5.1.1 Fields

FIELD	DATA TYPE	DESCRIPTION
Block Type	unint32	Initiates an Intrusion Event data block. This value is always 25.
Block Length	unint32	Total number of bytes in the Intrusion Event data block, including eight bytes for the Intrusion Event block type and length fields, plus the number of bytes of data that follows.
Device ID	unit32	Contains the identification number of the detecting managed device. You can obtain the managed device name by requesting Version 3 or 4 metadata. See Managed Device Record Metadata on page 99 for more information.
Event ID	uint32	Event identification number.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) of the event's detection.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment of the timestamp of the event's detection.
Rule ID (Signature ID)	uint32	Rule identification number that corresponds with the event.
Generator ID	uint32	Identification number of the Sourcefire 3D System preprocessor that generated the event.
Rule Revision	uint32	Rule revision number.
Classification ID	uint32	Identification number of the event classification message.

Intrusion Event Record 5.1.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Priority ID	uint32	Identification number of the priority associated with the event.
Source IP Address	uint8[16]	Source IPv4 or IPv6 address used in the event.
Destination IP Address	uint8[16]	Destination IPv4 or IPv6 address used in the event.
Source Port/ ICMP Type	uint16	The source port number if the event protocol type is TCP or UDP, or the ICMP type if the event is caused by ICMP traffic.
Destination Port/ICMP Code	uint16	The destination port number if the event protocol type is TCP or UDP, or the ICMP code if the event is caused by ICMP traffic.
IP Protocol Number	uint8	IANA-specified protocol number. For example: • 0 — IP • 1 — ICMP • 6 — TCP • 17 — UDP and so on.

Intrusion Event Record 5.1.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red. The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`
Impact	uint8	Impact flag value of the event. Values are: • `1` — Red (vulnerable) • `2` — Orange (potentially vulnerable) • `3` — Yellow (currently not vulnerable) • `4` — Blue (unknown target) • `5` — Gray (unknown impact)

Intrusion Event Record 5.1.1 Fields (Continued)

Field	Data Type	Description
Blocked	uint8	Value indicating whether the event was blocked. • `0` — not blocked • `1` — blocked • `2` — would be blocked (but not permitted by configuration)
MPLS Label	uint32	MPLS label.
VLAN ID	uint16	Indicates the ID of the VLAN where the packet originated.
Pad	uint16	Reserved for future use.
Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the intrusion policy.
User ID	uint32	The internal identification number for the user, if applicable.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.
Access Control Rule ID	uint32	A rule ID number that acts as a unique identifier for the access control rule.
Access Control Policy UUID	uint8[16]	A policy ID number that acts as a unique identifier for the access control policy.
Ingress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the ingress interface.
Egress Interface UUID	uint8[16]	An interface ID number that acts as a unique identifier for the egress interface.

Intrusion Event Record 5.1.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Ingress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the ingress security zone.
Egress Security Zone UUID	uint8[16]	A zone ID number that acts as a unique identifier for the egress security zone.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the connection event associated with the intrusion event.
Connection Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the connection event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.

Legacy Malware Event Data Structures

- Malware Event Data Block 5.1 on page 492
- Malware Event Data Block 5.1.1.x on page 497
- Malware Event Data Block 5.2.x on page 505

Malware Event Data Block 5.1

The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 16 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 1 and an event code of 101.

The following graphic shows the structure of the malware event data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Malware Event Block Type (16)</td></tr>
<tr><td></td><td colspan="32">Malware Event Block Length</td></tr>
</table>

	Agent UUID			
	Agent UUID, continued			
	Agent UUID, continued			
	Agent UUID, continued			
	Cloud UUID			
	Cloud UUID, continued			
	Cloud UUID, continued			
	Cloud UUID, continued			
	Timestamp			
	Event Type ID			
	Event Subtype ID	Host IP Address		
Detection Name	Host IP Address, cont.	Detector ID	String Block Type (0)	
	String Block Type (0), cont.		String Block Length	
	String Block Length, cont.		Detection Name...	
User	String Block Type (0)			
	String Block Length			
	User...			
File Name	String Block Type (0)			
	String Block Length			
	File Name...			
File Path	String Block Type (0)			
	String Block Length			
	File Path...			
File SHA Hash	String Block Type (0)			
	String Block Length			
	File SHA Hash...			
	File Size			
	File Type	File Timestamp		

Parent File Name	File Timestamp, cont.	String Block Type (0)
	String Block Type (0), cont.	String Block Length
	String Block Length, cont.	Parent File Name...
Parent File SHA Hash	String Block Type (0)	
	String Block Length	
	Parent File SHA Hash...	
Event Description	String Block Type (0)	
	String Block Length	
	Event Description...	

The Malware Event Data Block Fields table describes the fields in the malware event data block.

Malware Event Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always 16.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the FireAMP agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint8	The internal ID of the action that led to malware detection.

Malware Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Host IP Address	uint32	The host IP address associated with the malware event.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Sourcefire Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.

Malware Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint8	The file type of the detected or quarantined file.
File Timestamp	uint32	The creation timestamp of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.

Malware Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always 0.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.

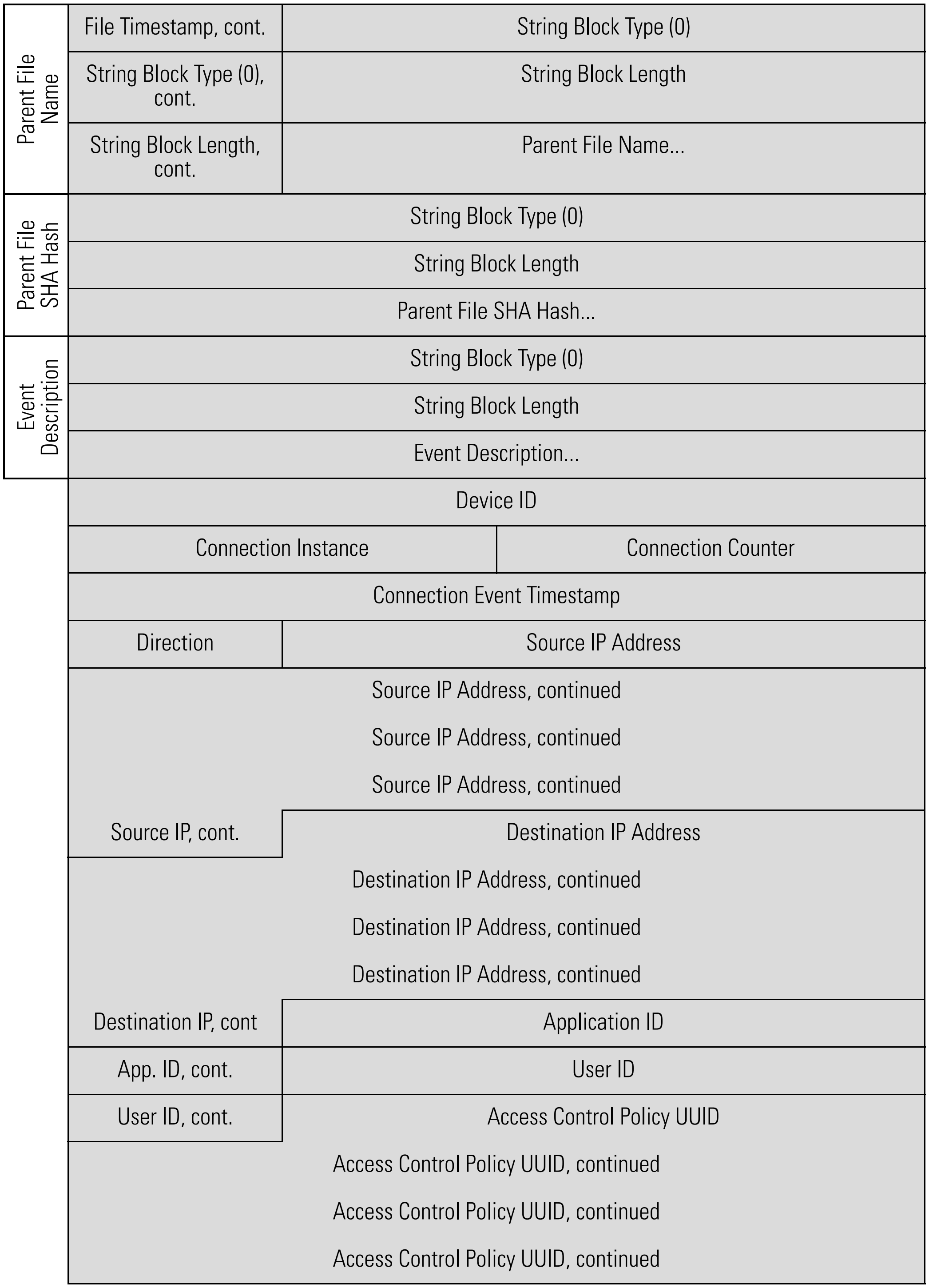
Malware Event Data Block 5.1.1.x

The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 24 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 2 and an event code of 101.

The following graphic shows the structure of the malware event data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Malware Event Block Type (24)																															
	Malware Event Block Length																															

<table>
<tr><td></td><td colspan="4">Agent UUID</td></tr>
<tr><td></td><td colspan="4">Agent UUID, continued</td></tr>
<tr><td></td><td colspan="4">Agent UUID, continued</td></tr>
<tr><td></td><td colspan="4">Agent UUID, continued</td></tr>
<tr><td></td><td colspan="4">Cloud UUID</td></tr>
<tr><td></td><td colspan="4">Cloud UUID, continued</td></tr>
<tr><td></td><td colspan="4">Cloud UUID, continued</td></tr>
<tr><td></td><td colspan="4">Cloud UUID, continued</td></tr>
<tr><td></td><td colspan="4">Malware Event Timestamp</td></tr>
<tr><td></td><td colspan="4">Event Type ID</td></tr>
<tr><td></td><td>Event Subtype ID</td><td colspan="3">Host IP Address</td></tr>
<tr><td rowspan="3">Detection Name</td><td>Host IP Address, cont.</td><td>Detector ID</td><td colspan="2">String Block Type (0)</td></tr>
<tr><td colspan="2">String Block Type (0), cont.</td><td colspan="2">String Block Length</td></tr>
<tr><td colspan="2">String Block Length, cont.</td><td colspan="2">Detection Name...</td></tr>
<tr><td rowspan="3">User</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">User...</td></tr>
<tr><td rowspan="3">File Name</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">File Name...</td></tr>
<tr><td rowspan="3">File Path</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">File Path...</td></tr>
<tr><td rowspan="3">File SHA Hash</td><td colspan="4">String Block Type (0)</td></tr>
<tr><td colspan="4">String Block Length</td></tr>
<tr><td colspan="4">File SHA Hash...</td></tr>
<tr><td></td><td colspan="4">File Size</td></tr>
<tr><td></td><td>File Type</td><td colspan="3">File Timestamp</td></tr>
</table>



Parent File Name	File Timestamp, cont.	String Block Type (0)
	String Block Type (0), cont.	String Block Length
	String Block Length, cont.	Parent File Name...
Parent File SHA Hash	String Block Type (0)	
	String Block Length	
	Parent File SHA Hash...	
Event Description	String Block Type (0)	
	String Block Length	
	Event Description...	
	Device ID	
	Connection Instance	Connection Counter
	Connection Event Timestamp	
	Direction	Source IP Address
	Source IP Address, continued	
	Source IP Address, continued	
	Source IP Address, continued	
	Source IP, cont.	Destination IP Address
	Destination IP Address, continued	
	Destination IP Address, continued	
	Destination IP Address, continued	
	Destination IP, cont	Application ID
	App. ID, cont.	User ID
	User ID, cont.	Access Control Policy UUID
	Access Control Policy UUID, continued	
	Access Control Policy UUID, continued	
	Access Control Policy UUID, continued	

URI	AC Pol UUID, cont.	Disposition	Retro. Disposition	Str. Block Type (0)
	String Block Type (0), continued			String Block Length
	String Block Length, continued			URI...
	Source Port		Destination Port	

The Malware Event Data Block for 5.1.1.x Fields table describes the fields in the malware event data block.

Malware Event Data Block for 5.1.1.x Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always 24.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the FireAMP agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint8	The internal ID of the action that led to malware detection.
Host IP Address	uint32	The host IP address associated with the malware event.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.

Malware Event Data Block for 5.1.1.x Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Sourcefire Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.

Malware Event Data Block for 5.1.1.x Fields (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint8	The file type of the detected or quarantined file.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.

Malware Event Data Block for 5.1.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always 0.
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • `1` — Download • `2` — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.

Malware Event Data Block for 5.1.1.x Fields (Continued)

Field	Data Type	Description
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — UNKNOWN — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Sourcefire cloud for a disposition. • 5 — NO_CLOUD_RESP — The Sourcefire cloud services did not respond to the request.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.

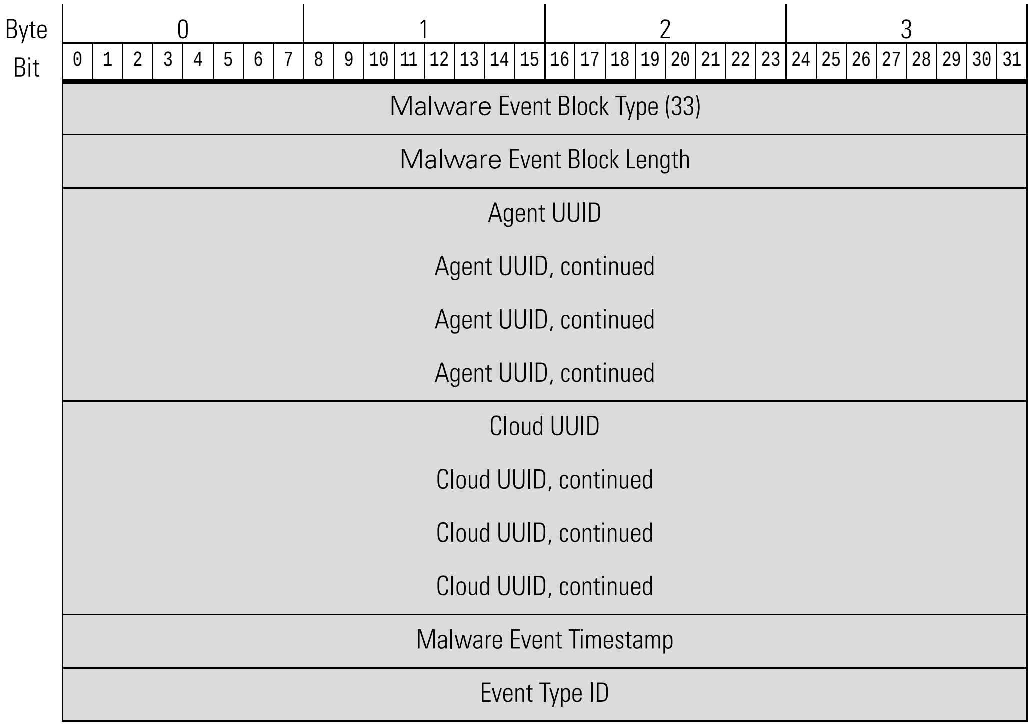
Malware Event Data Block for 5.1.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.

Malware Event Data Block 5.2.x

The eStreamer service uses the malware event data block to store information on malware events. These events contain information on malware detected or quarantined within a cloud, the detection method, and hosts and users affected by the malware. The malware event data block has a block type of 33 in the series 2 group of blocks. You request the event as part of the malware event record by setting the malware event flag—bit 30 in the request flags field—in the request message with an event version of 3 and an event code of 101.

The following graphic shows the structure of the malware event data block:



Detection Name	Event Subtype ID	Detector ID	String Block Type (0)	
	String Block Type (0), cont.		String Block Length	
	String Block Length, cont.		Detection Name...	
User	String Block Type (0)			
	String Block Length			
	User...			
File Name	String Block Type (0)			
	String Block Length			
	File Name...			
File Path	String Block Type (0)			
	String Block Length			
	File Path...			
File SHA Hash	String Block Type (0)			
	String Block Length			
	File SHA Hash...			
	File Size			
	File Type			
	File Timestamp			
Parent File Name	String Block Type (0)			
	String Block Length			
	Parent File Name...			
Parent File SHA Hash	String Block Type (0)			
	String Block Length			
	Parent File SHA Hash...			
Event Description	String Block Type (0)			
	String Block Length			
	Event Description...			
	Device ID			

<table>
<tr><td></td><td colspan="2">Connection Instance</td><td colspan="2">Connection Counter</td></tr>
<tr><td></td><td colspan="4">Connection Event Timestamp</td></tr>
<tr><td></td><td>Direction</td><td colspan="3">Source IP Address</td></tr>
<tr><td></td><td colspan="4">Source IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Source IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Source IP Address, continued</td></tr>
<tr><td></td><td>Source IP, cont.</td><td colspan="3">Destination IP Address</td></tr>
<tr><td></td><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td></td><td>Destination IP, cont</td><td colspan="3">Application ID</td></tr>
<tr><td></td><td>App. ID, cont.</td><td colspan="3">User ID</td></tr>
<tr><td></td><td>User ID, cont.</td><td colspan="3">Access Control Policy UUID</td></tr>
<tr><td></td><td colspan="4">Access Control Policy UUID, continued</td></tr>
<tr><td></td><td colspan="4">Access Control Policy UUID, continued</td></tr>
<tr><td></td><td colspan="4">Access Control Policy UUID, continued</td></tr>
<tr><td rowspan="3">URI</td><td>AC Pol UUID, cont.</td><td>Disposition</td><td>Retro. Disposition</td><td>Str. Block Type (0)</td></tr>
<tr><td colspan="3">String Block Type (0), continued</td><td>String Block Length</td></tr>
<tr><td colspan="3">String Block Length, continued</td><td>URI...</td></tr>
<tr><td></td><td colspan="2">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td></td><td colspan="2">Source Country</td><td colspan="2">Destination Country</td></tr>
<tr><td></td><td colspan="4">Web Application ID</td></tr>
<tr><td></td><td colspan="4">Client Application ID</td></tr>
<tr><td></td><td>Action</td><td>Protocol</td><td></td><td></td></tr>
</table>

The Malware Event Data Block for 5.2.x Fields table describes the fields in the malware event data block.

Malware Event Data Block for 5.2.x Fields

Field	Data Type	Description
Malware Event Block Type	uint32	Initiates a malware event data block. This value is always 33.
Malware Event Block Length	uint32	Total number of bytes in the malware event data block, including eight bytes for the malware event block type and length fields, plus the number of bytes of data that follows.
Agent UUID	uint8[16]	The internal unique ID of the FireAMP agent reporting the malware event.
Cloud UUID	uint8[16]	The internal unique ID of the malware awareness network from which the malware event originated.
Malware Event Timestamp	uint32	The malware event generation timestamp.
Event Type ID	uint32	The internal ID of the malware event type.
Event Subtype ID	uint8	The internal ID of the action that led to malware detection.
Detector ID	uint8	The internal ID of the detection technology that detected the malware.
String Block Type	uint32	Initiates a String data block containing the detection name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Detection Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Detection Name field.
Detection Name	string	The name of the detected or quarantined malware.
String Block Type	uint32	Initiates a String data block containing the username. This value is always 0.

Malware Event Data Block for 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	The number of bytes included in the User String data block, including eight bytes for the block type and header fields plus the number of bytes in the User field.
User	string	The user of the computer where the Sourcefire Agent is installed and where the malware event occurred. Note that these users are not tied to user discovery.
String Block Type	uint32	Initiates a String data block containing the file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Name field.
File Name	string	The name of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file path. This value is always 0.
String Block Length	uint32	The number of bytes included in the File Path String data block, including eight bytes for the block type and header fields plus the number of bytes in the File Path field.
File Path	string	The file path, not including the file name, of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the File SHA Hash field.

Malware Event Data Block for 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
File SHA Hash	string	The rendered string of the SHA-256 hash value of the detected or quarantined file.
File Size	uint32	The size in bytes of the detected or quarantined file.
File Type	uint8	The file type of the detected or quarantined file.
File Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the creation of the detected or quarantined file.
String Block Type	uint32	Initiates a String data block containing the parent file name. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File Name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File Name field.
Parent File Name	string	The name of the file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the parent file SHA hash. This value is always 0.
String Block Length	uint32	The number of bytes included in the Parent File SHA Hash String data block, including eight bytes for the block type and header fields plus the number of bytes in the Parent File SHA Hash field.
Parent File SHA Hash	string	The SHA-256 hash value of the parent file accessing the detected or quarantined file when detection occurred.
String Block Type	uint32	Initiates a String data block containing the event description. This value is always 0.

Malware Event Data Block for 5.2.x Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the Event Description String data block, including eight bytes for the block type and header fields plus the number of bytes in the Event Description field.
Event Description	string	The additional event information associated with the event type.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or IDS event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Event Timestamp	uint32	Timestamp of the connection event.
Direction	uint8	Indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.

Malware Event Data Block for 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Access Control Policy UUID	uint8[16]	Identification number that acts as a unique identifier for the access control policy that triggered the event.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — NEUTRAL — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Sourcefire cloud for a disposition, or the Sourcefire cloud services did not respond to the request.
Retrospective Disposition	uint8	Disposition of the file if the disposition is updated. If the disposition is not updated, this field contains the same value as the Disposition field. The possible values are the same as the Disposition field.
String Block Type	uint32	Initiates a String data block containing the URI. This value is always 0.
String Block Length	uint32	The number of bytes included in the URI data block, including eight bytes for the block type and header fields plus the number of bytes in the URI field.
URI	string	URI of the connection.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint 16	Code for the country of the destination host.

Malware Event Data Block for 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.

Legacy Discovery Data Structures

- Legacy Discovery Event Header on page 514
- Legacy Server Data Blocks on page 516
- Legacy Client Application Data Blocks on page 537
- Legacy Scan Result Data Blocks on page 543
- Legacy Vulnerability Blocks on page 563
- Legacy Host Profile Data Blocks on page 567
- Legacy OS Fingerprint Data Blocks on page 575

Legacy Discovery Event Header

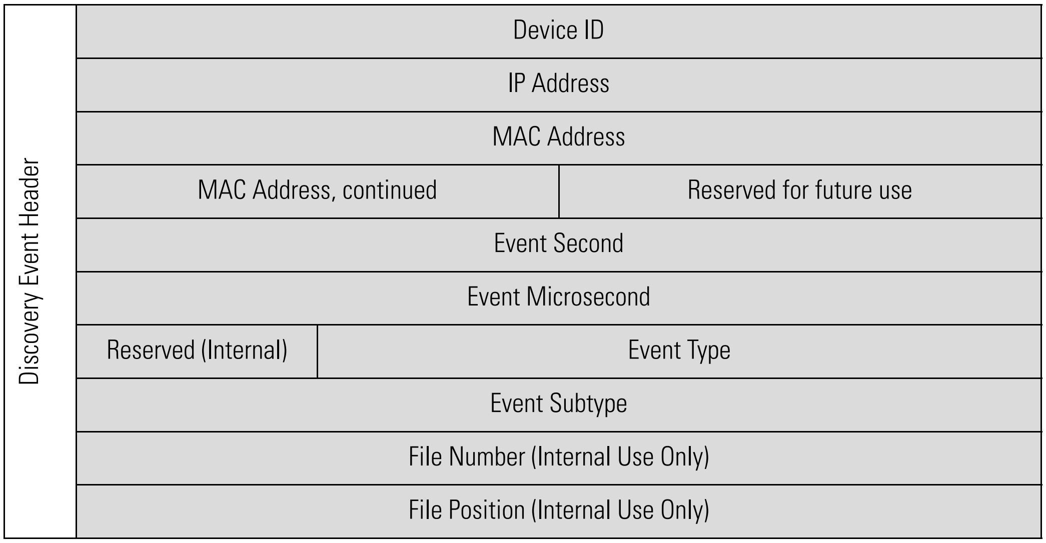
Discovery Event Header 4.8.0.2-5.1.1.x

Discovery and connection event messages contain a discovery event header. It conveys the type and subtype of the event, the time the event occurred, the device on which the event occurred, and the structure of the event data in the message. This header is followed by the actual host discovery, user, or connection event data. The structures associated with the different event type/ subtype values are described in Host Discovery Structures by Event Type on page 205.

The event type and event subtype fields of the discovery event header identify the structure of the transmitted event message. Once the structure of the event data block is determined, your program can parse the message appropriately.

The shaded rows in the following diagram illustrate the format of the discovery event header.

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="16">Header Version (1)</td><td colspan="16">Message Type (4)</td></tr>
<tr><td></td><td colspan="32">Message Length</td></tr>
<tr><td></td><td colspan="32">Record Type</td></tr>
<tr><td></td><td colspan="32">Record Length</td></tr>
<tr><td></td><td colspan="32">eStreamer Server Timestamp (in events, only if bit 23 is set)</td></tr>
<tr><td></td><td colspan="32">Reserved for Future Use (in events, only if bit 23 is set)</td></tr>
</table>



Discovery Event Header

Device ID

IP Address

MAC Address

MAC Address, continued | Reserved for future use

Event Second

Event Microsecond

Reserved (Internal) | Event Type

Event Subtype

File Number (Internal Use Only)

File Position (Internal Use Only)

The Discovery Event Header Fields table describes the discovery event header.

Discovery Event Header Fields

FIELD	DATA TYPES	DESCRIPTION
Device ID	uint32	ID number of the device that generated the discovery event. You can obtain the metadata for the device by requesting Version 3 and 4 metadata. See Managed Device Record Metadata on page 99 for more information.
IP Address	uint32	IP address of the host involved in the event.
MAC Address	uint8[6]	MAC address of the host involved in the event.
Reserved for future use	byte[2]	Two bytes of padding with values set to 0.
Event Second	uint32	UNIX timestamp (seconds since 01/01/1970) that the system generated the event.
Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the system generated the event.
Reserved (Internal)	byte	Internal data from Sourcefire and can be disregarded.

Discovery Event Header Fields (Continued)

FIELD	DATA TYPES	DESCRIPTION
Event Type	uint32	Event type (1000 for new events, 1001 for change events, 1002 for user input events, 1050 for full host profile). See Host Discovery Structures by Event Type on page 205 for a list of available event types.
Event Subtype	uint32	Event subtype. See Host Discovery Structures by Event Type on page 205 for a list of available event subtypes.
File Number	byte[4]	Serial file number. This field is for Sourcefire internal use and can be disregarded.
File Position	byte[4]	Event's position in the serial file. This field is for Sourcefire internal use and can be disregarded.

Legacy Server Data Blocks

For more information, see the following sections:

- Host Server Data Block for Version 4.9.0.x on page 516
- Web Application Data Block for 4.9.1 - 4.10.x on page 519
- Host Server Data Block for 4.9.1.x on page 520
- Full Server Data Block for 4.9.0.x on page 523
- Full Server Data Block for 4.9.1.x on page 529
- Server Information Data Block for 4.9.1 and Earlier on page 534
- Attribute Address Data Block for 4.5.x - 5.1.1.x on page 536

Host Server Data Block for Version 4.9.0.x

The Host Server data block conveys information about servers identified by the system, including the server port, the frequency of use, last use, and confidence, as well as lists of server information blocks and sub-server blocks for the host for the event. Host Server data blocks are contained in messages for new TCP and UDP servers and changes to TCP and UDP servers.

Server data for this data block for 4.9.0.x is encapsulated in lists of server information blocks rather than through individual fields, allowing for multiple servers.

The Host Server data block has a block type of 89.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Host Server data block:

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		Server Block Type (89)			
		Server Block Length			
		Port		Hits	
		Hits, continued		Last Used	
Server Information Blocks		Last Used, cont.		Generic List Block Type (31)	
		Generic List Block Type (31)		Generic List Block Length	
		Generic List Block Length		Server Information Data Blocks...	
		Server Information Data Blocks...			
		List Block Type (11)			
		List Block Length			
Svc Subtype		Sub-Server Block Type (1) *			
		Sub-Serve Block Length			
		Sub-Serve Data...			
		Confidence			

The Host Server Data 4.9.0.x Fields table describes the fields of the Host Server data block:

Host Server Data 4.9.0.x Fields

FIELD	DATA TYPE	DESCRIPTION
Host Server Block Type	uint32	Initiates a Host Server data block. This value is always 89.
Host Server Block Length	uint32	Total number of bytes in the Host Server data block, including the eight bytes in the Host Server block type and length fields plus the number of bytes of data that follows.
Port	uint16	Port number on which the server runs.
Hits	uint32	Number of hits the server has received.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the server in use.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields plus the number of bytes in all of the encapsulated data blocks.
Server Information Data Blocks	variable	Encapsulated Server Information data blocks up to the maximum number of bytes in the list block length.
List Block Type	uint32	Initiates a list of Sub-Server data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including eight bytes for the list block type and length fields plus the number of bytes in the encapsulated Sub-Server data blocks that follow.
Sub-Server Block Type	uint32	Initiates the first Sub-Server data block. This data block can be followed by other Sub-Server data blocks up to the limit defined in the list block length field.

Host Server Data 4.9.0.x Fields (Continued)

Field	Data Type	Description
Sub-Server Block Length	uint32	Total number of bytes in each Sub-Server data block, including the eight bytes in the Sub-Server block type and length fields plus the number of bytes of data that follows.
Sub-Server Data	variable	Sub-server data as documented in Sub-Server Data Block on page 241.
Confidence	uint32	System confidence percentage.

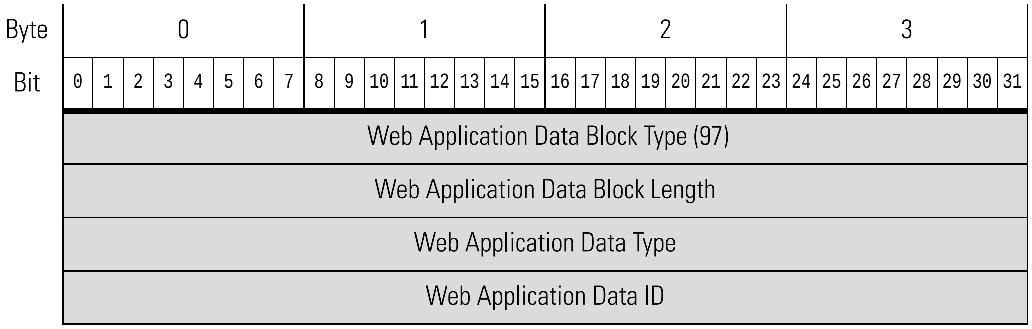
Web Application Data Block for 4.9.1 - 4.10.x

The web application data block has a block type of 97. Identity data blocks are used in Host Server, Full Server, Host Client Application, and Connection Statistics data blocks. The data block describes the web application type and application ID from HTTP client requests detected by the system.

For more information on the data blocks that incorporate this data block, see the following sections:

- Host Server Data Block for Version 4.9.0.x on page 516
- Full Server Data Block for 4.9.0.x on page 523
- Host Client Application Data Block for 3.5 - 4.9.0.x on page 538
- Connection Statistics Data Block for 4.7 - 4.9.0.x on page 577

The following diagram shows the format of a Web Application data block 4.9.1+.



The Web Application Data Block Fields table describes the fields of the Web Application data block:

Web Application Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Web Application Data Block Type	uint32	Initiates the Web Application data block. This value is always 97.
Web Application Data Block Length	uint32	Number of bytes in the Web Application data block. This value should always be sixteen bytes for the data block type and length fields and the source type and ID fields.
Web Application ID	uint32	Indicates the ID of the web application.

Host Server Data Block for 4.9.1.x

The Host Server data block conveys information about servers identified by the system, including the server port, the frequency of use, last use, and confidence, as well as lists of server information blocks and Sub-Server blocks for the host for the event. Host Server data blocks are contained in messages for new TCP and UDP servers and changes to TCP and UDP servers. For more information, see Server Messages on page 206. Starting in 4.9.1, the data block includes a list of Web Application data blocks. Note that the Host Server data block has a block type of 98.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Host Server data block:

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Server Block Type (98)				
	Server Block Length				
	Port		Hits		
	Hits, continued		Last Used		
Server Information Blocks	Last Used, cont.		Generic List Block Type (31)		
	Generic List Block Type (31)		Generic List Block Length		
	Generic List Block Length		Server Information Data Blocks...		
	Server Information Data Blocks...				
	List Block Type (11)				Scan Subtype List
	List Block Length				
Svc Subtype	Sub-Server Block Type (1) *				
	Sub-Server Block Length				
	Sub-Server Data...				
	Confidence				
Web App's	Generic List Block Type (31)				
	Generic List Block Length				
	Web Application Data...				

The Host Server Data 4.9.0.x Fields table describes the fields of the Host Server data block:

Host Server Data Fields 4.9.1.x

FIELD	DATA TYPE	DESCRIPTION
Host Server Block Type	uint32	Initiates a Host Server data block. This value is always 98.
Host Server Block Length	uint32	Total number of bytes in the Host Server data block, including the eight bytes in the Host Server block type and length fields, plus the number of bytes of data that follows.
Port	uint16	Port number where the server runs.
Hits	uint32	Number of hits the server has received.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the server in use.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated data blocks.
Server Information Data Blocks	variable	Encapsulated Server Information data blocks up to the maximum number of bytes in the list block length.
List Block Type	uint32	Initiates a list of Sub-Server data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including eight bytes for the list block type and length fields, plus the number of bytes in the encapsulated Sub-Server data blocks that follow.
Sub-Server Block Type	uint32	Initiates the first Sub-Server data block. This data block can be followed by other Sub-Server data blocks up to the limit defined in the list block length field.

Host Server Data Fields 4.9.1.x (Continued)

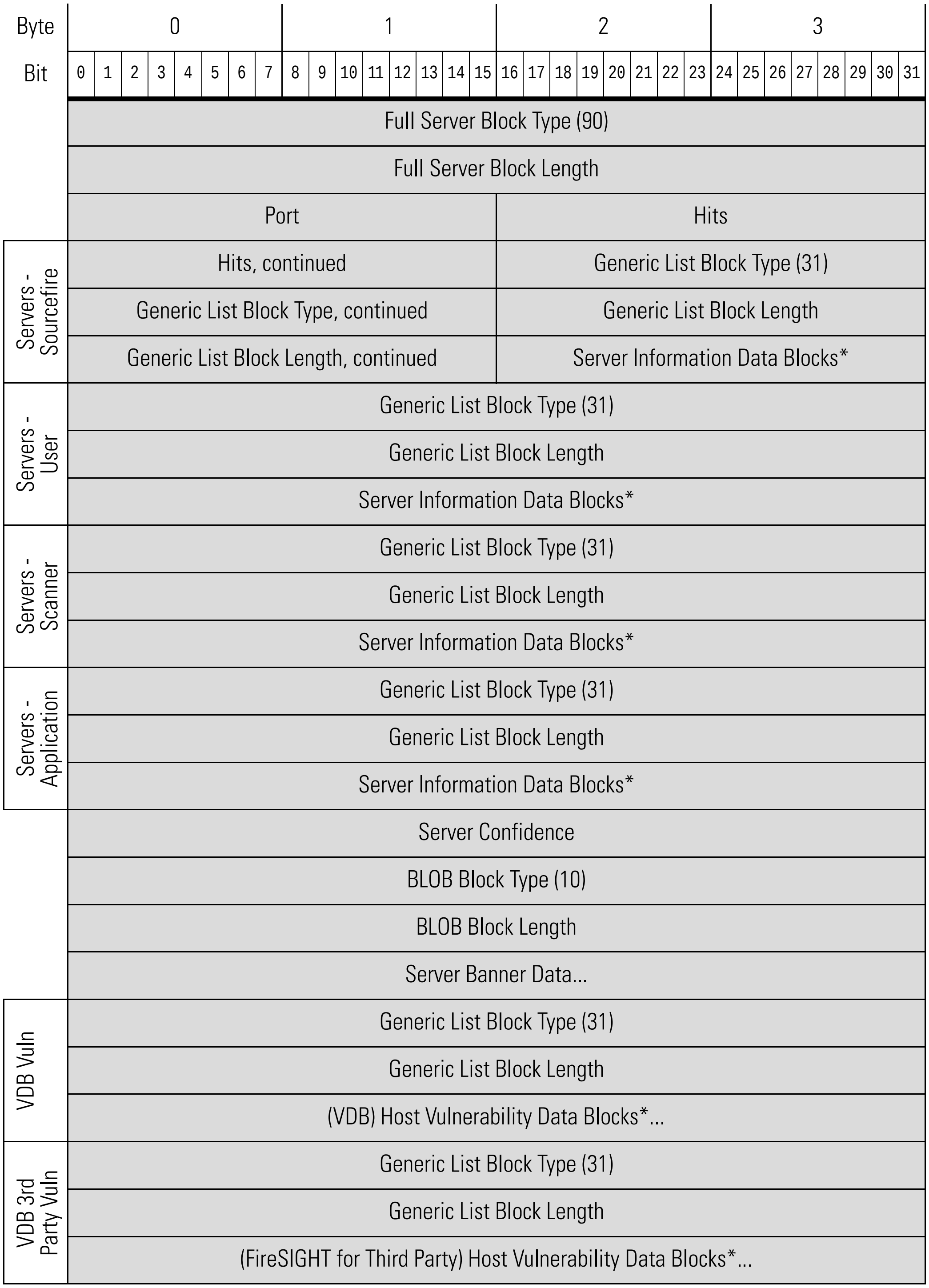
FIELD	DATA TYPE	DESCRIPTION
Sub-Server Block Length	uint32	Total number of bytes in each Sub-Server data block, including the eight bytes in the Sub-Server block type and length fields, plus the number of bytes of data that follows.
Sub-Server Data	variable	Sub-server data as documented in Sub-Server Data Block on page 241.
Confidence	uint32	System confidence percentage.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated Web Application data blocks.
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length.

Full Server Data Block for 4.9.0.x

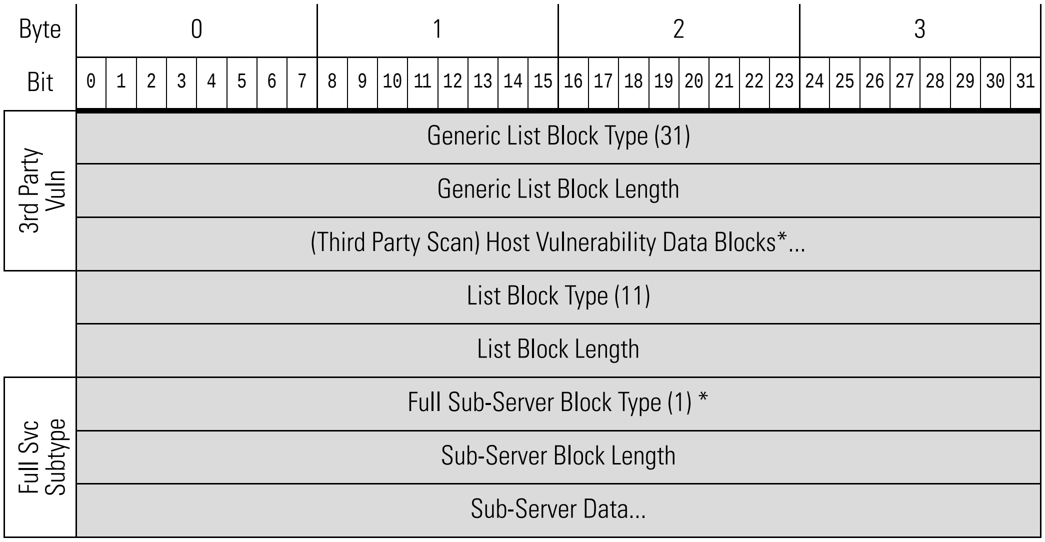
The Full Server data block conveys information about a server, including the server port, the frequency of use and most recent update, server ID, vendor, product, and version, confidence of data accuracy, Sourcefire and third-party vulnerabilities related to that server for the host for the event, and source type and source identification. A Full Server data block for each TCP and UDP server on the host in the event is included in a list in the Full Host Profile data block. Changes for the 4.9.0.x data block include new source type and source ID fields and a 32-bit server ID field. The Full Server data block has a block type of 90.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Server data block:



	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		Full Server Block Type (90)			
		Full Server Block Length			
		Port		Hits	
Servers - Sourcefire		Hits, continued		Generic List Block Type (31)	
		Generic List Block Type, continued		Generic List Block Length	
		Generic List Block Length, continued		Server Information Data Blocks*	
Servers - User		Generic List Block Type (31)			
		Generic List Block Length			
		Server Information Data Blocks*			
Servers - Scanner		Generic List Block Type (31)			
		Generic List Block Length			
		Server Information Data Blocks*			
Servers - Application		Generic List Block Type (31)			
		Generic List Block Length			
		Server Information Data Blocks*			
		Server Confidence			
		BLOB Block Type (10)			
		BLOB Block Length			
		Server Banner Data...			
VDB Vuln		Generic List Block Type (31)			
		Generic List Block Length			
		(VDB) Host Vulnerability Data Blocks*...			
VDB 3rd Party Vuln		Generic List Block Type (31)			
		Generic List Block Length			
		(FireSIGHT for Third Party) Host Vulnerability Data Blocks*...			



	Byte 0 (Bits 0–7)	Byte 1 (Bits 8–15)	Byte 2 (Bits 16–23)	Byte 3 (Bits 24–31)
3rd Party Vuln	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party Scan) Host Vulnerability Data Blocks*...			
	List Block Type (11)			
	List Block Length			
Full Svc Subtype	Full Sub-Server Block Type (1) *			
	Sub-Server Block Length			
	Sub-Server Data...			

The Full Server Data Block 4.9.0.x Fields table describes the components of the Full Server data block.

Full Server Data Block 4.9.0.x Fields

FIELD	DATA TYPE	DESCRIPTION
Full Server Block Type	uint32	Initiates a Full Server data block. This value is always 90.
Full Server Block Length	uint32	Total number of bytes in the Full Server data block, including eight bytes for the full server block type and length fields plus the number of bytes of full server data that follows.
Port	uint16	Server port number.
Hits	uint32	Number of hits the server has received.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by the system. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.

Full Server Data Block 4.9.0.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host identified by the system. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host added by a user. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by a scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host added by a scanner. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.

Full Server Data Block 4.9.0.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host added by an application. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Server Confidence	uint32	Percentage of confidence of the system in its correct identification of the server.
BLOB Block Type	uint32	Initiates the BLOB data block, that contains banner data. This value is always 10.
BLOB Block Length	uint32	Total number of bytes in the BLOB data block, including eight bytes for the block type and length fields plus the number of bytes in the banner.
Server Banner Data	byte[*N*]	First *N* bytes of the packet involved in the server event, where *N* is equal to or less than 256.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB host vulnerability data for vulnerabilities identified by a third party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities identified by Sourcefire. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying host vulnerability data generated by a third party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.

Full Server Data Block 4.9.0.x Fields (Continued)

Field	Data Type	Description
Third Party Scan (VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about VDB vulnerability data for vulnerabilities identified by a third party scanner. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party host vulnerability data generated by a third party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third Party Scan Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing the original third party vulnerability data for vulnerabilities identified by a third party scanner. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Sub-Server data blocks conveying server subtype data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields plus all encapsulated Server data blocks. This field is followed by zero or more Full Sub-Server data blocks.
Full Sub-Server Block Type	uint32	Initiates the first Full Sub-Server data block. This data block can be followed by other Full Sub-Server data blocks up to the limit defined in the list block length field.
Full Sub-Server Block Length	uint32	Total number of bytes in each Full Sub-Server data block, including the eight bytes in the Full Sub-Server block type and length fields plus the number of bytes of data that follows.
Full Sub-Server Data Blocks *	uint32	Full Sub-Server data blocks containing sub-server information for the server. See Full Server Data Block for 4.9.1.x on page 529 for a description of this data block.

Full Server Data Block for 4.9.1.x

The Full Server data block conveys information about a server, including the server port, the frequency of use and most recent update, server ID, vendor, product, and version, confidence of data accuracy, Sourcefire and third-party vulnerabilities related to that server for the host for the event, and source type and source identification. A Full Server data block for each TCP and UDP server on the host in the event is included in a list in the Full Host Profile data block. The 4.9.1+ data block includes a new list of Web Application data blocks. The Full Server data block has a block type of 99.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the Full Server data block:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Full Server Block Type (99)</td></tr>
<tr><td></td><td colspan="32">Full Server Block Length</td></tr>
<tr><td></td><td colspan="16">Port</td><td colspan="16">Hits</td></tr>
<tr><td rowspan="3">Servers - VDB</td><td colspan="16">Hits, continued</td><td colspan="16">Generic List Block Type (31)</td></tr>
<tr><td colspan="16">Generic List Block Type, continued</td><td colspan="16">Generic List Block Length</td></tr>
<tr><td colspan="16">Generic List Block Length, continued</td><td colspan="16">Server Information Data Blocks*</td></tr>
<tr><td rowspan="3">Servers - User</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Server Information Data Blocks*</td></tr>
<tr><td rowspan="3">Servers - Scanner</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Server Information Data Blocks*</td></tr>
<tr><td rowspan="3">Servers - Application</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">Server Information Data Blocks*</td></tr>
<tr><td></td><td colspan="32">Server Confidence</td></tr>
</table>

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	BLOB Block Type (10)			
	BLOB Block Length			
	Server Banner Data...			
VDB Vuln	Generic List Block Type (31)			
	Generic List Block Length			
	(VDB) Host Vulnerability Data Blocks*...			
VDB 3rd Party Vuln	Generic List Block Type (31)			
	Generic List Block Length			
	(FireSIGHT for Third Party) Host Vulnerability Data Blocks*...			
3rd Party Vuln	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party Scan) Host Vulnerability Data Blocks*...			
	List Block Type (11)			
	List Block Length			
Full Svc Subtype	Full Sub-Server Block Type (51) *			
	Sub-Server Block Length			
	Sub-Server Data...			
Web App's	Web Application Block Type (97)*			
	Web Application Block Length			
	Web Application Data...			

The Full Server Data Block 4.9.0.x Fields table describes the components of the Full Server data block.

Full Server Data Block 4.9.1.x Fields

Field	Data Type	Description
Full Server Block Type	uint32	Initiates a Full Server data block. This value is always 99.
Full Server Block Length	uint32	Total number of bytes in the Full Server data block, including eight bytes for the full server block type and length fields, plus the number of bytes of full server data that follows.
Port	uint16	Server port number.
Hits	uint32	Number of hits the server has received.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by the system. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Server Information Data Blocks *	variable	Server information data blocks containing information about servers identified on a host. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host added by a user. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by a scanner. This value is always 31.

Full Server Data Block 4.9.1.x Fields (Continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host added by a scanner. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising server information data blocks conveying server data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated server information data blocks.
Server Information Data Blocks *	variable	Server information data blocks containing information about servers on a host added by an application. See Server Information Data Block for 4.9.1 and Earlier on page 534 for a description of this data block.
Server Confidence	uint32	Percentage of confidence of the system in its correct identification of the server.
BLOB Block Type	uint32	Initiates the BLOB data block, that contains banner data. This value is always 10.
BLOB Block Length	uint32	Total number of bytes in the BLOB data block, including eight bytes for the block type and length fields, plus the number of bytes in the banner.
Server Banner Data	byte[*N*]	First *N* bytes of the packet involved in the server event, where *N* is equal to or less than 256.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB host vulnerability data for vulnerabilities identified by a third party scanner. This value is always 31.

Full Server Data Block 4.9.1.x Fields (Continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about host vulnerabilities identified by Sourcefire. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party host vulnerability data generated by a third party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third Party Scan (VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing information about VDB vulnerability data for vulnerabilities identified by a third party scanner. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying vulnerability data generated by a third party scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Host Vulnerability data blocks.
Third Party Scan Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks containing the original third party vulnerability data for vulnerabilities identified by a third party scanner. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server Subtype data blocks conveying sub-server data. This value is always 11.

Full Server Data Block 4.9.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Full Sub-Server data blocks.
Full Sub-Server Block Type	uint32	Initiates the first Full Sub-Server data block. This data block can be followed by other Full Sub-Server data blocks up to the limit defined in the list block length field.
Full Sub-Server Block Length	uint32	Total number of bytes in each Sub-Server data block, including the eight bytes in the Full Sub-Server block type and length fields, plus the number of bytes of data that follows.
Full Sub-Server Data Blocks *	variable	Full Sub-Server data blocks containing sub-servers for the server. See Full Server Data Block for 4.9.1.x on page 529 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated Web Application data blocks.
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length.

Server Information Data Block for 4.9.1 and Earlier

The Server Information data block conveys information about a server, including the server ID, server vendor and version, and source information. The Server Information data block has a block type of 88. Server information data blocks are conveyed in lists within host server and full server data blocks. For more information see Host Server Data Block for Version 4.9.0.x on page 516 and Full Server Data Block for 4.9.0.x on page 523.

The following diagram shows the format of the Server Information data block:

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Server Information Block Type (88)																															
	Server Information Block Length																															
	Server ID																															
	String Block Type (0)																															
	String Block Length																															
	Server Vendor Name String...																															
	String Block Type (0)																															
	String Block Length																															
	Server Version String...																															
	Last Used																															
	Source Type																															
	Source ID																															

The Server Information Data Block 4.9.1 and Earlier Fields table describes the components of the Server Information data block.

Server Information Data Block 4.9.1 and Earlier Fields

FIELD	DATA TYPE	DESCRIPTION
Server Information Block Type	uint32	Initiates a Server Information data block. This value is always 88.
Server Information Block Length	uint32	Total number of bytes in the Server Information data block, including eight bytes for the Server Information block type and length fields, four bytes for the server ID, eight bytes for the vendor name block type and length, another four for the vendor name, eight bytes for the version string block type and length, another four for the version string, and four bytes each for the last used, source type, and source ID fields.

Server Information Data Block 4.9.1 and Earlier Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Server ID	uint32	Indicates the ID of the server identified in the data block.
String Block Type	uint32	Initiates a String data block containing the server vendor’s name. This value is always 0.
String Block Length	uint32	Number of bytes in the vendor name String data block, including eight bytes for the block type and length fields, plus the number of bytes in the server vendor name.
Server Vendor Name	string	Name of the server vendor.
String Block Type	uint32	Initiates a String data block that contains the server version. This value is always 0.
String Block Length	uint32	Number of bytes in the server version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the server version.
Server Version	string	Server version.
Last Time Used	uint32	Indicates when the server information was last used in traffic.
Source Type	uint32	Indicates the type (Sourcefire, user, application, or scanner) of the source that supplied the server information.
Source ID	uint32	Indicates the ID of the source that supplied the server information.

Attribute Address Data Block for 4.5.x - 5.1.1.x

The Attribute Address data block contains an attribute list item and is used within an Attribute Definition data block. It has a block type of 38.

The following diagram shows the basic structure of an Attribute Address data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Attribute Address Block Type (38)			
	Attribute Address Block Length			
	Attribute ID			
	IP Address			
	Bits			

The Attribute Address Data Block Fields table describes the fields of the Attribute Address data block.

Attribute Address Data Block Fields

Field	Data Type	Description
Attribute Address Block Type	uint32	Initiates an Attribute Address data block. This value is always 38.
Attribute Address Block Length	uint32	Number of bytes in the Attribute Address data block, including eight bytes for the attribute address block type and length, plus the number of bytes in the attribute address data that follows.
Attribute ID	uint32	Identification number of the affected attribute, if applicable.
IP Address	uint8[4]	IP address of the host, if the address was automatically assigned, in IP address octets.
Bits	uint32	Contains the significant bits used to calculate the netmask if an IP address was automatically assigned.

Legacy Client Application Data Blocks

For more information, see the following sections:

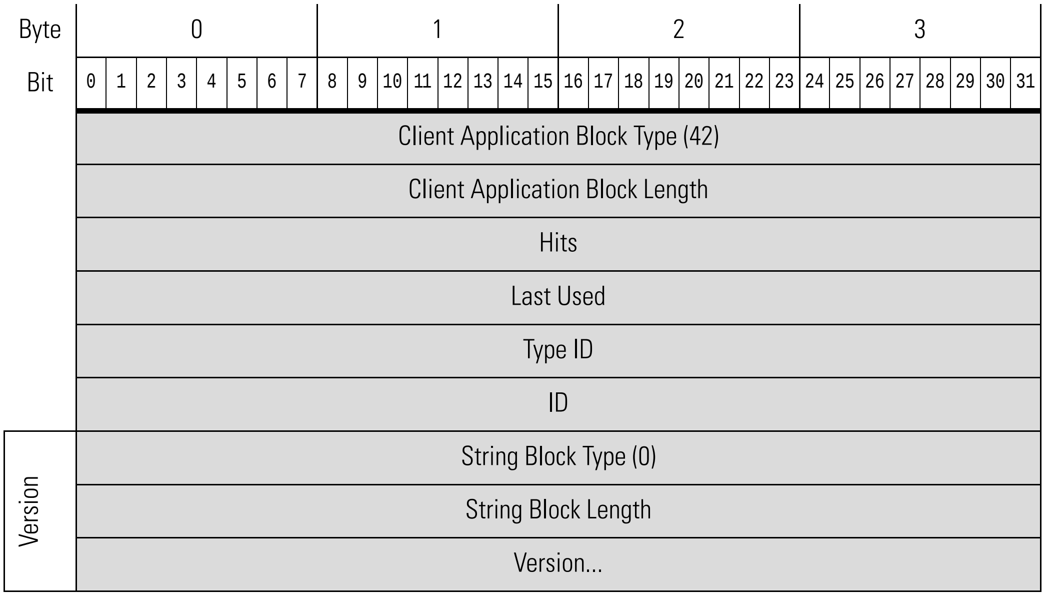
- Host Client Application Data Block for 3.5 - 4.9.0.x on page 538
- Host Client Application Data Block for 4.9.1 - 4.10.x on page 539

- User Client Application Data Block for 5.1 and earlier on page 541

Host Client Application Data Block for 3.5 - 4.9.0.x

The Client Application data block for 3.5 - 4.9.0.x describes a client application and is used within legacy New Client Application events (event type 1001, subtype 7) and Client Application Timeout events (event type 1001, subtype 20). It has a block type of 42.

The following diagram shows the basic structure of a Client Application data block:



The Client Application Data Block 3.5 - 4.9.0.x Fields table describes the fields of the Client Application data block.

Client Application Data Block 3.5 - 4.9.0.x Fields

FIELD	DATA TYPE	DESCRIPTION
Client Application Block Type	uint32	Initiates a Host Client Application data block. This value is always 42.
Client Application Block Length	uint32	Number of bytes in the Client Application data block, including eight bytes for the client application block type and length, plus the number of bytes in the client application data that follows.

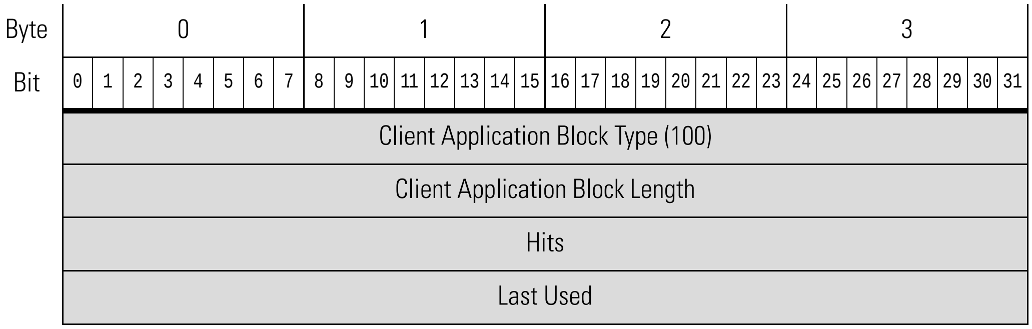
Client Application Data Block 3.5 - 4.9.0.x Fields (Continued)

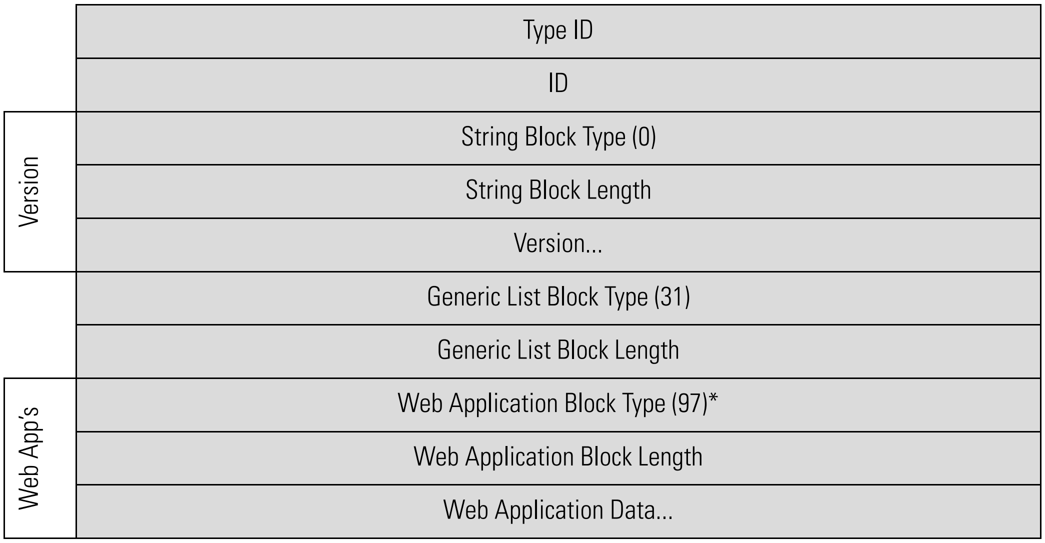
FIELD	DATA TYPE	DESCRIPTION
Hits	uint32	Number of times the system has detected the client application in use.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the client in use.
Type ID	uint32	Identification number of the detected client application type, if applicable.
ID	uint32	Identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application name, including eight bytes for the string block type and length plus the number of bytes in the client application version.
Version	string	Client application version.

Host Client Application Data Block for 4.9.1 - 4.10.x

The Client Application data block for 4.9.1 - 4.10.x describes a client application and is used within New Client Application events (event type 1001, subtype 7) and Client Application Timeout events (event type 1001, subtype 20). The Client Application data block for 4.9.1 - 4.10.x has a block type of 100. Its successor, introduced for 5.0+, has a block type of 122.

The following diagram shows the basic structure of a Client Application data block:




Type ID
ID
Version
String Block Type (0)
String Block Length
Version...
Generic List Block Type (31)
Generic List Block Length
Web App's
Web Application Block Type (97)*
Web Application Block Length
Web Application Data...

The Client Application Data Block for 4.9.1 - 4.10.x Fields table describes the fields of the Client Application data block.

Client Application Data Block for 4.9.1 - 4.10.x Fields

FIELD	DATA TYPE	DESCRIPTION
Client Application Block Type	uint32	Initiates a Host Client Application data block. This value is always 100.
Client Application Block Length	uint32	Number of bytes in the Client Application data block, including eight bytes for the client application block type and length, plus the number of bytes in the client application data that follows.
Hits	uint32	Number of times the system has detected the client application in use.
Last Used	uint32	UNIX timestamp that represents the last time the system detected the client in use.
Type ID	uint32	Identification number of the detected client application type, if applicable.
ID	uint32	Identification number of the detected client application, if applicable.

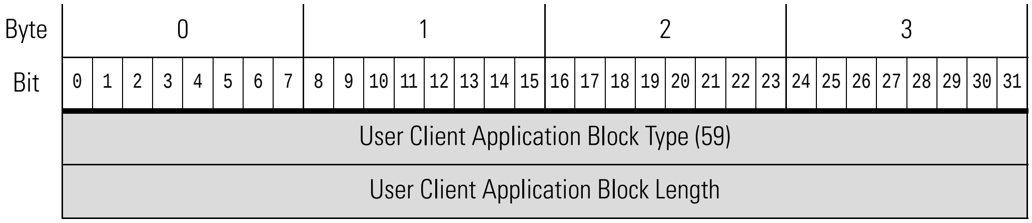
Client Application Data Block for 4.9.1 - 4.10.x Fields (Continued)

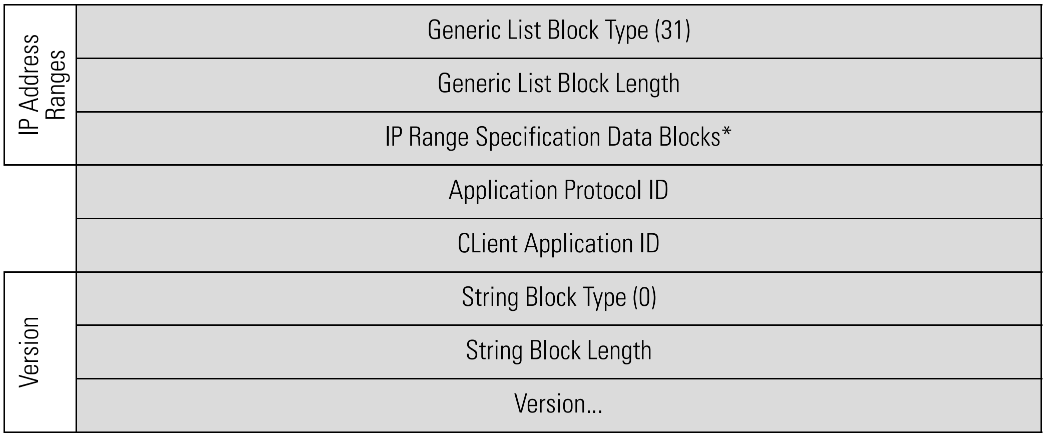
FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application name, including eight bytes for the string block type and length, plus the number of bytes in the client application version.
Version	string	Client application version.
Generic List Block Type	uint32	Initiates a Generic List data block. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List block and encapsulated Web Application data blocks. This number includes the eight bytes of the generic list block header fields, plus the number of bytes in all of the encapsulated Web Application data blocks.
Web Application Data Blocks	variable	Encapsulated Web Application data blocks up to the maximum number of bytes in the list block length. For information on the encapsulated Web Application data blocks, see Web Application Data Block for 4.9.1 - 4.10.x on page 519.

User Client Application Data Block for 5.1 and earlier

The User Client Application data block contains information about the source of the client application data, the identification number for the user who added the data, and the lists of IP address range data blocks. The User Client Application data block has a block type of 59.

The following diagram shows the basic structure of a User Client Application data block:





The User Client Application Data Block Fields table describes the fields of the User Client Application data block.

User Client Application Data Block Fields

FIELD	NUMBER OF BYTES	DESCRIPTION
User Client Application Block Type	uint32	Initiates a User Client Application data block. This value is always 59.
User Client Application Block Length	uint32	Total number of bytes in the User Client Application data block, including eight bytes for the user client application block type and length fields, plus the number of bytes of user client application data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See User Server Data Block Fields on page 280 for a description of this data block.
Application Protocol ID	uint32	The internal identification number for the application protocol, if applicable.

User Client Application Data Block Fields (Continued)

Field	Number of Bytes	Description
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block that contains the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the client application version String data block, including the string block type and length fields, plus the number of bytes in the version.
Version	string	Client application version.

Legacy Scan Result Data Blocks

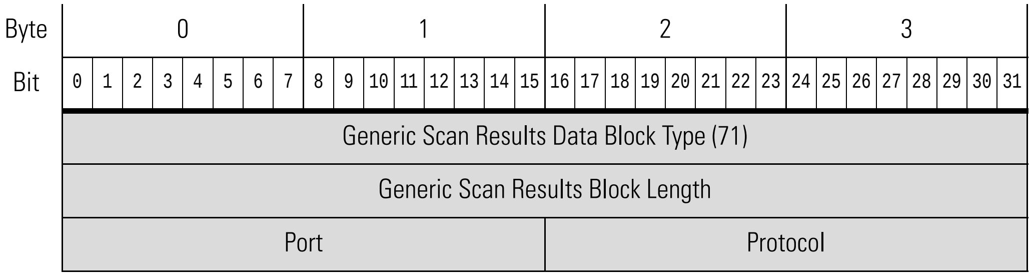
For more information, see the following sections:

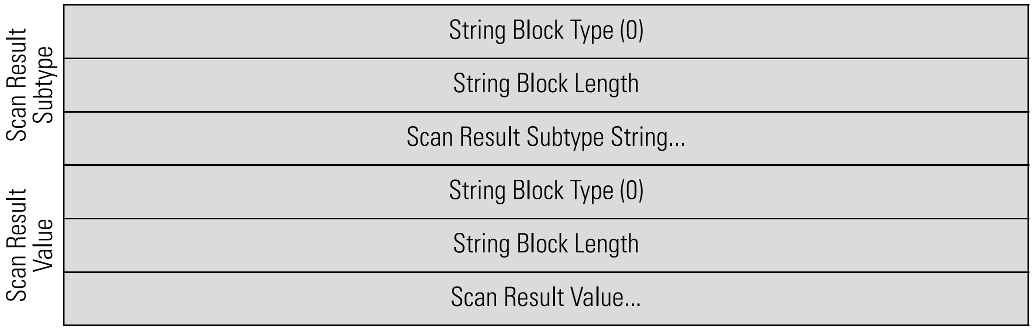
- Generic Scan Results Data Block for 4.9.1.x and earlier on page 543
- Scan Result Data Block for 4.6.1 - 4.9.1.x on page 545
- Scan Result Data Block 4.10.0 - 5.1.1.x on page 548
- Scan Vulnerability Data Block for 4.9 - 4.9.1.x on page 551
- User Product Data Block for 4.10.x, 5.0 - 5.0.x on page 554

Generic Scan Results Data Block for 4.9.1.x and earlier

The Generic Scan Results data block contains scan results and is used in the Scan Result Data Block for 4.6.1 - 4.9.1.x on page 545. The Generic Scan Results data block has a block type of 71.

The following diagram shows the basic structure of a Generic Scan Results data block:





Scan Result Subtype	String Block Type (0)
	String Block Length
	Scan Result Subtype String...
Scan Result Value	String Block Type (0)
	String Block Length
	Scan Result Value...

The Generic Scan Result Data Block for 4.9.1.x and earlier Fields table describes the fields of the Generic Scan Results data block.

Generic Scan Result Data Block for 4.9.1.x and earlier Fields

Field	Number of Bytes	Description
Generic Scan Results Data Block Type	uint32	Initiates a Generic Scan Results data block. This value is always 71.
Generic Scan Results Block Length	uint32	Total number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes of scan results data that follows.
Port	uint16	Port used by the sub-server affected by the vulnerabilities in the results.
Protocol	uint16	Network protocol. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP and so on.
String Block Type	uint32	Initiates a String data block that contains the sub-server. This value is always 0.
String Block Length	uint32	Number of bytes in the sub-server String data block, including eight bytes for the block type and length fields, plus the number of bytes in the sub-server.

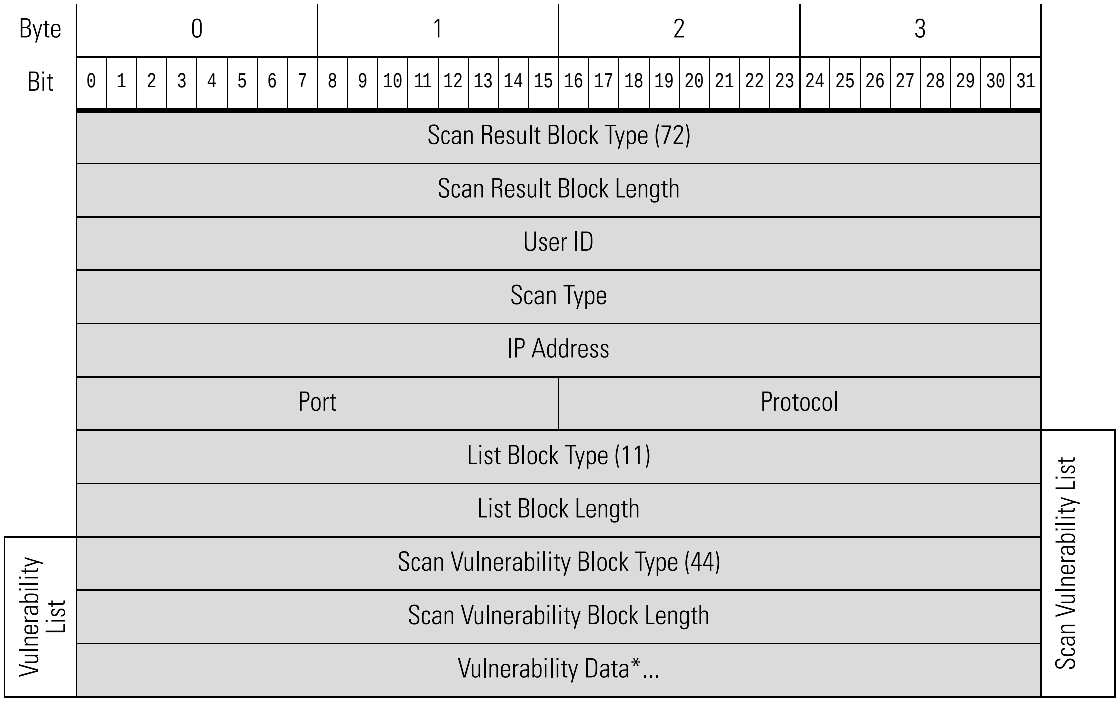
Generic Scan Result Data Block for 4.9.1.x and earlier Fields (Continued)

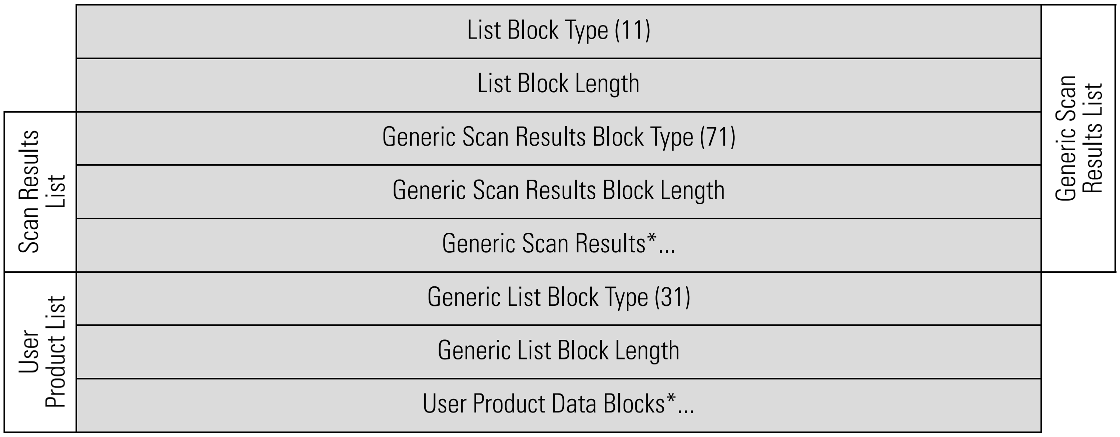
FIELD	NUMBER OF BYTES	DESCRIPTION
Scan Result Subtype	string	Scan result subtype.
String Block Type	uint32	Initiates a String data block that contains the value. This value is always 0.
String Block Length	uint32	Number of bytes in the value String data block, including eight bytes for the block type and length fields, plus the number of bytes in the value.
Scan result value	string	Scan result value.

Scan Result Data Block for 4.6.1 - 4.9.1.x

The Scan Result data block describes a vulnerability and is used within Add Scan Result events (event type 1002, subtype 11). The Scan Result data block has a block type of 72.

The following diagram shows the format of a Scan Result data block:





	List Block Type (11)	Generic Scan Results List
	List Block Length	
Scan Results List	Generic Scan Results Block Type (71)	
	Generic Scan Results Block Length	
	Generic Scan Results*...	
User Product List	Generic List Block Type (31)	
	Generic List Block Length	
	User Product Data Blocks*...	

The Scan Result Data Block for 4.6.1 - 4.9.1.x Fields table describes the fields of the Scan Result data block.

Scan Result Data Block for 4.6.1 - 4.9.1.x Fields

Field	Data Type	Description
Scan Result Block Type	uint32	Initiates a Scan Result data block. This value is always 72.
Scan Result Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
User ID	uint32	Contains the user identification number for the user who imported the scan result or ran the scan that produced the scan result.
Scan Type	uint32	Indicates how the results were added to the sensor. Values include: • Nessus — 1 • Nmap — 2
IP Address	uint32	IP address of the host affected by the vulnerabilities in the result, in IP address octets.
Port	uint16	Port used by the sub-server affected by the vulnerabilities in the results.

Scan Result Data Block for 4.6.1 - 4.9.1.x Fields (Continued)

Field	Data Type	Description
Protocol	uint16	Network protocol. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP and so on.
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block describing a vulnerability detected during a scan. This value is always 44.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes in the scan vulnerability data that follows.
Vulnerability Data*	variable	Information relating to each vulnerability.
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport scan vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.

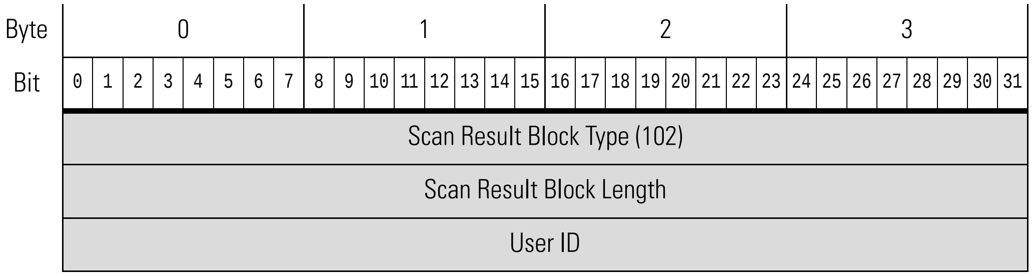
Scan Result Data Block for 4.6.1 - 4.9.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic Scan Results Block Type	uint32	Initiates a Generic Scan Results data block describing server and operating system data detected during a scan. This value is always 71.
Generic Scan Results Block Length	uint32	Number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes in the scan result data that follows.
Generic Scan Results Data*	variable	Information relating to each scan result.
Generic List Block Type	uint32	Initiates a Generic List data block comprising User Product data blocks conveying host input data from a third party application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated User Product data blocks.
User Product Data Blocks *	variable	User Product data blocks with a block type of 65 containing host input data. See User Product Data Block for 4.10.x, 5.0 - 5.0.x on page 554 for a description of this data block.

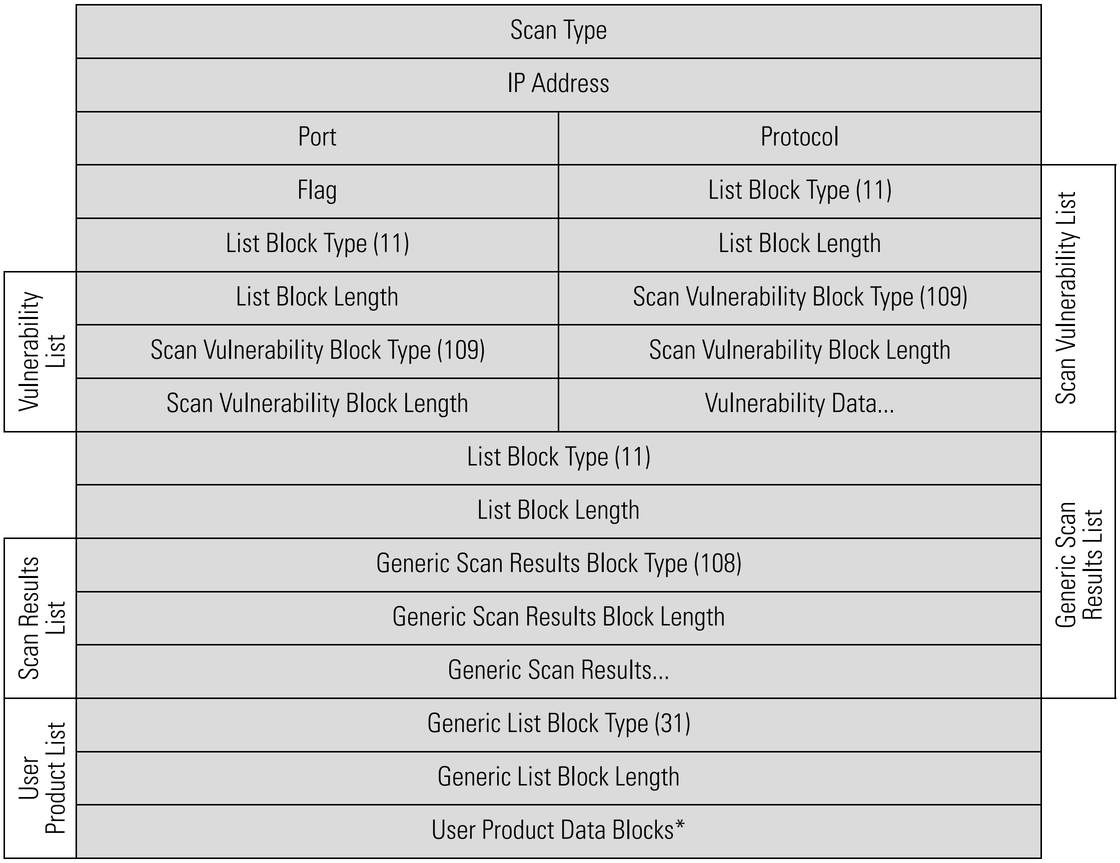
Scan Result Data Block 4.10.0 - 5.1.1.x

The Scan Result data block describes a vulnerability and is used within Add Scan Result events (event type 1002, subtype 11). The Scan Result data block has a block type of 102.

The following diagram shows the format of a Scan Result data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Scan Result Block Type (102)			
	Scan Result Block Length			
	User ID			



	Scan Type		
	IP Address		
	Port	Protocol	
	Flag	List Block Type (11)	Scan Vulnerability List
	List Block Type (11)	List Block Length	
Vulnerability List	List Block Length	Scan Vulnerability Block Type (109)	
	Scan Vulnerability Block Type (109)	Scan Vulnerability Block Length	
	Scan Vulnerability Block Length	Vulnerability Data...	
	List Block Type (11)		Generic Scan Results List
	List Block Length		
Scan Results List	Generic Scan Results Block Type (108)		
	Generic Scan Results Block Length		
	Generic Scan Results...		
User Product List	Generic List Block Type (31)		
	Generic List Block Length		
	User Product Data Blocks*		

The Scan Result Data Block Fields table describes the fields of the Scan Result data block.

Scan Result Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
Scan Result Block Type	uint32	Initiates a Scan Result data block. This value is always 102.
Scan Result Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
User ID	uint32	Contains the user identification number for the user who imported the scan result or ran the scan that produced the scan result.

Scan Result Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Scan Type	uint32	Indicates how the results were added to the system.
IP Address	uint32	IP address of the host affected by the vulnerabilities in the result, in IP address octets.
Port	uint16	Port used by the sub-server affected by the vulnerabilities in the results.
Protocol	uint16	IANA protocol number. For example: • `1` — ICMP • `4` — IP • `6` — TCP • `17` — UDP and so on.
Flag	uint16	Reserved
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block describing a vulnerability detected during a scan. This value is always 109.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes in the scan vulnerability data that follows.
Vulnerability Data	string	Information relating to each vulnerability.

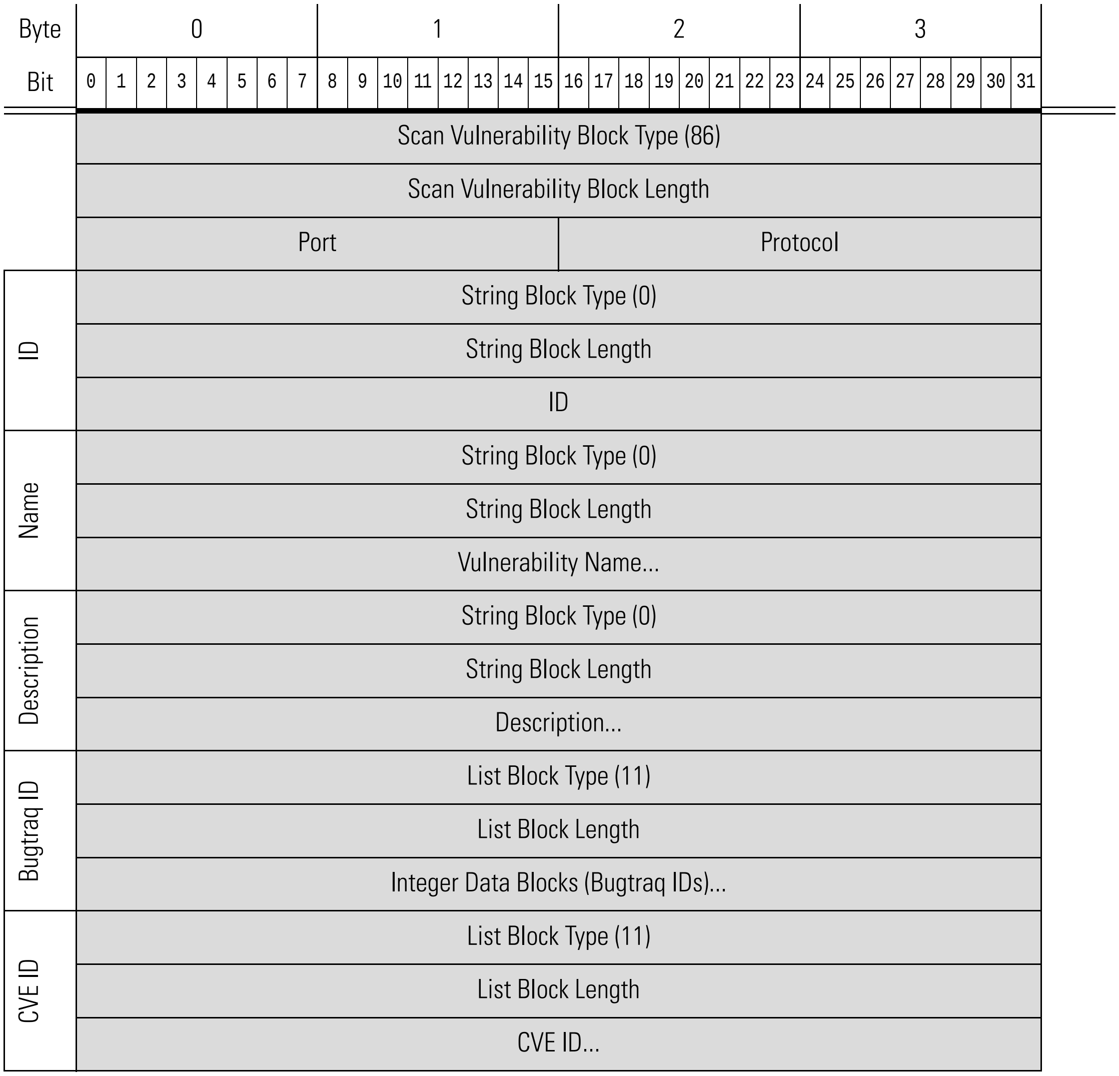
Scan Result Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Type	uint32	Initiates a List data block comprising Scan Vulnerability data blocks conveying transport Scan Vulnerability data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Scan Vulnerability data blocks. This field is followed by zero or more Scan Vulnerability data blocks.
Generic Scan Results Block Type	uint32	Initiates a Generic Scan Results data block describing server and operating system data detected during a scan. This value is always 108.
Generic Scan Results Block Length	uint32	Number of bytes in the Generic Scan Results data block, including eight bytes for the generic scan results block type and length fields, plus the number of bytes in the scan result data that follows.
Generic Scan Results Data	string	Information relating to each scan result.
Generic List Block Type	uint32	Initiates a Generic List data block comprising User Product data blocks conveying host input data from a third party application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated User Product data blocks.
User Product Data Blocks *	variable	User Product data blocks containing host input data. See User Product Data Block 5.1+ on page 353 for a description of this data block.

Scan Vulnerability Data Block for 4.9 - 4.9.1.x

The Scan Vulnerability data block describes a vulnerability and is used within Scan Result data blocks, that in turn are used in Add Scan Result events (event type 1002, subtype 11). For more information, see Scan Result Data Block for 4.6.1 - 4.9.1.x on page 545 and Add Scan Result Messages on page 221. The Scan Vulnerability data block has a block type of 86.

The following diagram shows the format of a Scan Vulnerability data block:



Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	Scan Vulnerability Block Type (86)																															
	Scan Vulnerability Block Length																															
	Port																Protocol															
ID	String Block Type (0)																															
	String Block Length																															
	ID																															
Name	String Block Type (0)																															
	String Block Length																															
	Vulnerability Name...																															
Description	String Block Type (0)																															
	String Block Length																															
	Description...																															
Bugtraq ID	List Block Type (11)																															
	List Block Length																															
	Integer Data Blocks (Bugtraq IDs)...																															
CVE ID	List Block Type (11)																															
	List Block Length																															
	CVE ID...																															

The Scan Vulnerability Data Block for 4.9 - 4.9.1.x Fields table describes the fields of the Scan Vulnerability data block.

Scan Vulnerability Data Block for 4.9 - 4.9.1.x Fields

FIELD	DATA TYPE	DESCRIPTION
Scan Vulnerability Block Type	uint32	Initiates a Scan Vulnerability data block. This value is always 86.
Scan Vulnerability Block Length	uint32	Number of bytes in the Scan Vulnerability data block, including eight bytes for the scan vulnerability block type and length fields, plus the number of bytes of scan vulnerability data that follows.
Port	uint16	Port used by the sub-server affected by the vulnerability.
Protocol	uint16	Network protocol. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP and so on.
String Block Type	uint32	Initiates a String data block for the ID.
String Block Length	uint32	Number of bytes in the String data block for the ID, including eight bytes for the string block type and length, plus the number of bytes in the ID.
ID	string	The ID for the reported vulnerability as specified by the scan utility that detected it. For a vulnerability detected by a Nessus scan, for example, this field indicates the Nessus ID.
String Block Type	uint32	Initiates a String data block for the vulnerability name.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Name	string	Name of the vulnerability.

Scan Vulnerability Data Block for 4.9 - 4.9.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block for the vulnerability description.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability description, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability description.
Description	string	Description of the vulnerability.
String Block Type	uint32	Initiates a String data block for the vulnerability name.
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Bugtraq ID	string	Contains zero or more Integer (INT32) data blocks that form a list of Bugtraq identification numbers.
List Block Type	uint32	Initiates a List data block for the list of Common Vulnerability Exposure (CVE) identification numbers.
List Block Length	uint32	Number of bytes in the List data block for the CVE identification number, including eight bytes for the string block type and length, plus the number of bytes in the CVE identification number.
CVE ID	string	Contains zero or more String Information data blocks that form a list of CVE identification numbers.

User Product Data Block for 4.10.x, 5.0 - 5.0.x

The User Product data block conveys host input data imported from a third party application, including third party application string mappings. This data block is used in Scan Result Data Block 5.2+ on page 308. The User Product data block has a block type of 65 for 4.10.x, and a block type of 118 for 5.0 - 5.0.x. The block types have the same structure.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

The following diagram shows the format of the User Product data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	User Product Data Block Type (65 | 118)			
	User Product Block Length			
	Source ID			
	Source Type			
IP Address Ranges	Generic List Block Type (31)			
	Generic List Block Length			
	IP Range Specification Data Blocks*			
	Port		Protocol	
	Drop User Product			
Custom Vendor String	String Block Type (0)			
	String Block Length			
	Custom Vendor String...			
Custom Product String	String Block Type (0)			
	String Block Length			
	Custom Product String...			
Custom Version String	String Block Type (0)			
	String Block Length			
	Custom Version String...			
	Software ID			
	Server ID			
	Vendor ID			
	Product ID			

Major Version String	String Block Type (0)
	String Block Length
	Major Version String...
Minor Version String	String Block Type (0)
	String Block Length
	Minor Version String...
Revision String	String Block Type (0)
	String Block Length
	Revision String...
To Major String	String Block Type (0)
	String Block Length
	To Major Version String...
To Minor String	String Block Type (0)
	String Block Length
	To Minor Version String...
To Revision String	String Block Type (0)
	String Block Length
	To Revision String...
Build String	String Block Type (0)
	String Block Length
	Build String...
Patch String	String Block Type (0)
	String Block Length
	Patch String...
Extension String	String Block Type (0)
	String Block Length
	Extension String...

OS UUID	Operating System UUID
	Operating System UUID cont.
	Operating System UUID cont.
	Operating System UUID cont.
List of Fixes	Generic List Block Type (31)
	Generic List Block Length
	Fix List Data Blocks*

The User Product Data Block Fields for 4.10.x, 5.0-5.0.x table describes the components of the User Product data block.

User Product Data Block Fields for 4.10.x, 5.0-5.0.x

FIELD	DATA TYPE	DESCRIPTION
User Product Data Block Type	uint32	Initiates a User Product data block. This value is 65 for version 4.10.x and 118 for version 5.0 - 5.0.x.
User Product Block Length	uint32	Total number of bytes in the User Product data block, including eight bytes for the user product block type and length fields, plus the number of bytes in the user product data that follows.
Source ID	uint32	Identification number of the source that imported the data.
Source Type	uint32	The source type of the source that supplied the data.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Port	uint16	Port specified by the user.

User Product Data Block Fields for 4.10.x, 5.0-5.0.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
Protocol	uint16	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP and so on.
Drop User Product	uint32	Indicates whether the user OS definition was deleted from the host: • 0 — No • 1 — Yes
String Block Type	uint32	Initiates a String data block containing the custom vendor name specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom vendor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the vendor name.
Custom Vendor Name	string	The custom vendor name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom product name specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom product String data block, including eight bytes for the block type and length fields, plus the number of bytes in the product name.
Custom Product Name	string	The custom product name specified in the user input.
String Block Type	uint32	Initiates a String data block containing the custom version specified in the user input. This value is always 0.
String Block Length	uint32	Number of bytes in the custom version String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.

User Product Data Block Fields for 4.10.x, 5.0-5.0.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
Custom Version	string	The custom version specified in the user input.
Software ID	uint32	The identifier for a specific revision of a server or operating system in the Sourcefire database.
Server ID	uint32	The Sourcefire application identifier for the application protocol on the host server specified in user input.
Vendor ID	uint32	The identifier for the vendor of a third party operating system specified when the third party operating system is mapped to a Sourcefire 3D operating system definition.
Product ID	uint32	The product identification string of a third party operating system string specified when the third party operating system string is mapped to a Sourcefire 3D operating system definition.
String Block Type	uint32	Initiates a String data block containing the major version number of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
Major Version	string	Major version of the Sourcefire 3D operating system definition that a third party operating system string is mapped to.
String Block Type	uint32	Initiates a String data block containing the minor version number of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.

User Product Data Block Fields for 4.10.x, 5.0-5.0.x (Continued)

Field	Data Type	Description
Minor Version	string	Minor version number of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the revision number of the Sourcefire operating system definition that a third party operating system string in the user input is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
Revision	string	Revision number of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last major version of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Major String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.
To Major	string	Last version number in a range of major version numbers of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the last minor version of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Minor String data block, including eight bytes for the block type and length fields, plus the number of bytes in the version.

User Product Data Block Fields for 4.10.x, 5.0-5.0.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
To Minor	string	Last version number in a range of minor version numbers of the Sourcefire 3D operating system definition that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the Last revision number of the Sourcefire 3D operating system definition that a third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the To Revision String data block, including eight bytes for the block type and length fields, plus the number of bytes in the revision number.
To Revision	string	Last revision number in a range of revision numbers of the Sourcefire 3D operating system definitions that a third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the build number of the Sourcefire 3D operating system that the third party operating system string is mapped. This value is always 0.
String Block Length	uint32	Number of bytes in the build String data block, including eight bytes for the block type and length fields, plus the number of bytes in the build number.
Build	string	Build number of the Sourcefire 3D operating system that the third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the patch number of the Sourcefire 3D operating system that the third party operating system string is mapped to. This value is always 0.
String Block Length	uint32	Number of bytes in the patch String data block, including eight bytes for the block type and length fields, plus the number of bytes in the patch number.

User Product Data Block Fields for 4.10.x, 5.0-5.0.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
Patch	string	Patch number of the Sourcefire 3D operating system that the third party operating system string in the user input is mapped to.
String Block Type	uint32	Initiates a String data block containing the extension number of the Sourcefire 3D operating system that the third party operating system string is mapped. This value is always 0.
String Block Length	uint32	Number of bytes in the extension String data block, including eight bytes for the block type and length fields, plus the number of bytes in the extension number.
Extension	string	Extension number of the Sourcefire 3D operating system that the third party operating system string in the user input is mapped to.
UUID	uint8 [x16]	Contains the unique identification number for the operating system.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Fix List data blocks conveying user input data regarding what fixes have been applied to hosts in the specified IP address ranges. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Fix List data blocks.
Fix List Data Blocks *	variable	Fix List data blocks containing information about fixes applied to the hosts. See Fix List Data Block on page 279 for a description of this data block.

Legacy Vulnerability Blocks

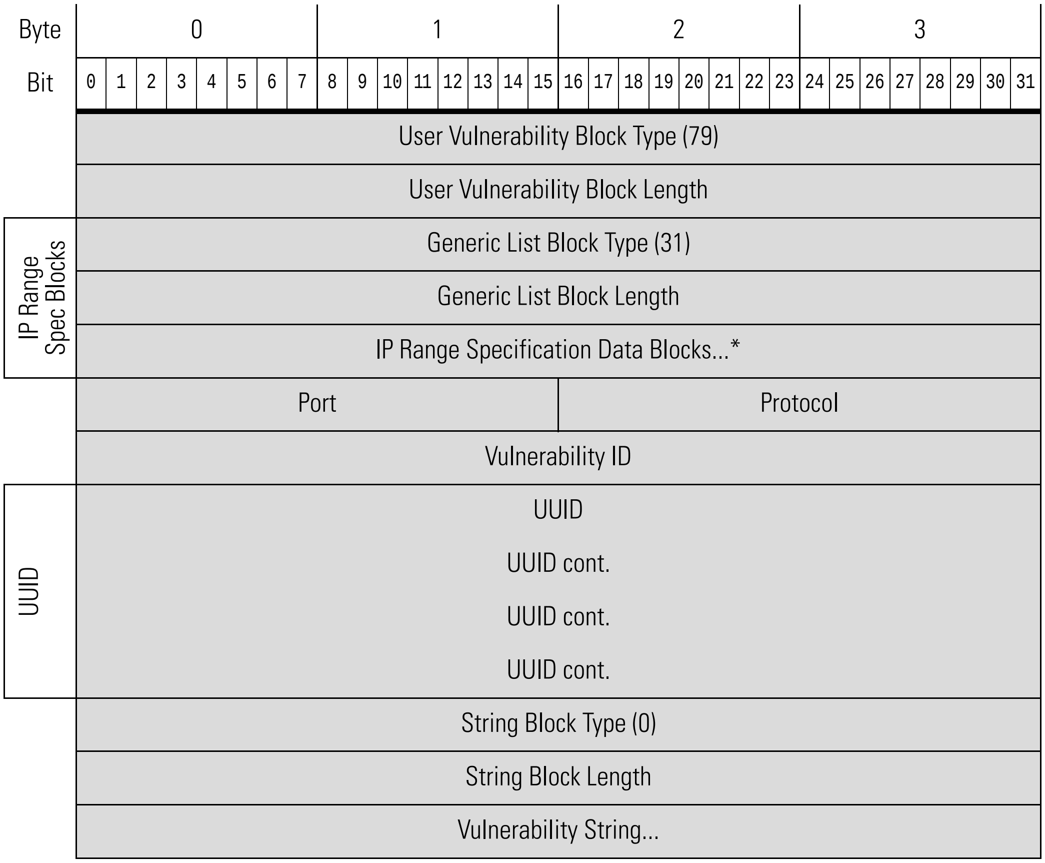
See the following sections for more information:

- User Vulnerability Data Block 4.7 - 4.10.x on page 563

User Vulnerability Data Block 4.7 - 4.10.x

The User Vulnerability data block describes a vulnerability and is used within User Vulnerability Change data blocks, which in turn are used in User Set Valid Vulnerabilities events (event type 1002, subtype 1) and User Set Invalid Vulnerabilities events (event type 1002, subtype 2). The User Vulnerability data block has a block type of 79. For more information on User Vulnerability Change data blocks, see User Vulnerability Change Data Block 4.7+ on page 285.

The following diagram shows the format of a User Vulnerability data block:



	Byte 0 (Bit 0–7)	Byte 1 (Bit 8–15)	Byte 2 (Bit 16–23)	Byte 3 (Bit 24–31)
	User Vulnerability Block Type (79)			
	User Vulnerability Block Length			
IP Range Spec Blocks	Generic List Block Type (31)			
	Generic List Block Length			
	IP Range Specification Data Blocks...*			
	Port		Protocol	
	Vulnerability ID			
UUID	UUID			
	UUID cont.			
	UUID cont.			
	UUID cont.			
	String Block Type (0)			
	String Block Length			
	Vulnerability String...			

The User Vulnerability Data Block Fields table describes the fields of the User Vulnerability data block:

User Vulnerability Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
User Vulnerability Block Type	uint32	Initiates a User Vulnerability data block. This value is always 22.
User Vulnerability Block Length	uint32	Number of bytes in the User Vulnerability data block, including eight bytes for the user vulnerability block type and length fields, plus the number of bytes of user vulnerability data that follows.
Generic List Block Type	uint32	Initiates a Generic List data block comprising IP Range Specification data blocks conveying IP address range data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated IP Range Specification data blocks.
IP Range Specification Data Blocks *	variable	IP Range Specification data blocks containing information about the IP address ranges for the user input. See IP Address Range Data Block for 5.2+ on page 270 for a description of this data block.
Port	uint16	Port used by the sub-server affected by the vulnerability.
Protocol	uint16	The IANA protocol number. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP and so on.
Vulnerability ID	uint32	Sourcefire vulnerability ID.
UUID	uint8 [16]	Contains the unique identification number for the vulnerability.
String Block Type	uint32	Initiates a String data block for the vulnerability name.

User Vulnerability Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the String data block for the vulnerability name, including eight bytes for the string block type and length, plus the number of bytes in the vulnerability name.
Vulnerability Name	string	Vulnerability name.

Legacy User Login Data Blocks

See the following sections for more information:

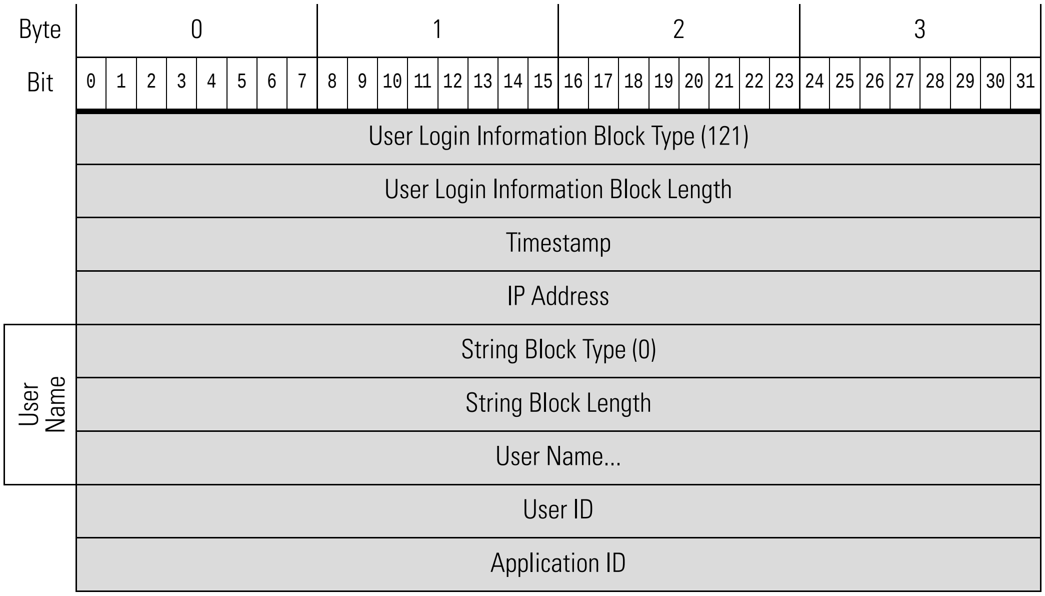
- User Login Information Data Block for 5.0 - 5.0.2 on page 565

User Login Information Data Block for 5.0 - 5.0.2

The User Login Information data block is used in User Information Update messages and conveys changes in login information for a detected user. For more information, see User Information Update Message Block on page 223.

The User Login Information data block has a block type of 121 for version 5.0 - 5.0.2.

The graphic below shows the format of the User Login Information data block:



	Byte 0 (Bit 0-7)	Byte 1 (Bit 8-15)	Byte 2 (Bit 16-23)	Byte 3 (Bit 24-31)
	User Login Information Block Type (121)			
	User Login Information Block Length			
	Timestamp			
	IP Address			
User Name	String Block Type (0)			
	String Block Length			
	User Name...			
	User ID			
	Application ID			

<table>
<tr><td></td><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td></td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td rowspan="3" colspan="2">Email</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">Email...</td></tr>
</table>

The User Login Information Data Block Fields 5.0 - 5.0.2 table describes the components of the User Login Information data block.

User Login Information Data Block Fields 5.0 - 5.0.2

Field	Data Type	Description
User Login Information Block Type	uint32	Initiates a User Login Information data block. This value is 121 for version 5.0 - 5.0.2.
User Login Information Block Length	uint32	Total number of bytes in the User Login Information data block, including eight bytes for the user login information block type and length fields, plus the number of bytes in the user login information data that follows.
Timestamp	uint32	Timestamp of the event.
IP Address	uint8[4]	IP address from the host where the user was detected logging in, in IP address octets.
String Block Type	uint32	Initiates a String data block containing the username for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the username String data block, including eight bytes for the block type and length fields, plus the number of bytes in the username.
Username	string	The user name for the user.
User ID	uint32	Identification number of the user.
Application ID	uint32	The application ID for the application protocol used in the connection that the login information was derived from.

User Login Information Data Block Fields 5.0 - 5.0.2 (Continued)

Field	Data Type	Description
String Block Type	uint32	Initiates a String data block containing the email address for the user. This value is always 0.
String Block Length	uint32	Number of bytes in the email address String data block, including eight bytes for the block type and length fields, plus the number of bytes in the email address.
Email	string	The email address for the user.

Legacy Host Profile Data Blocks

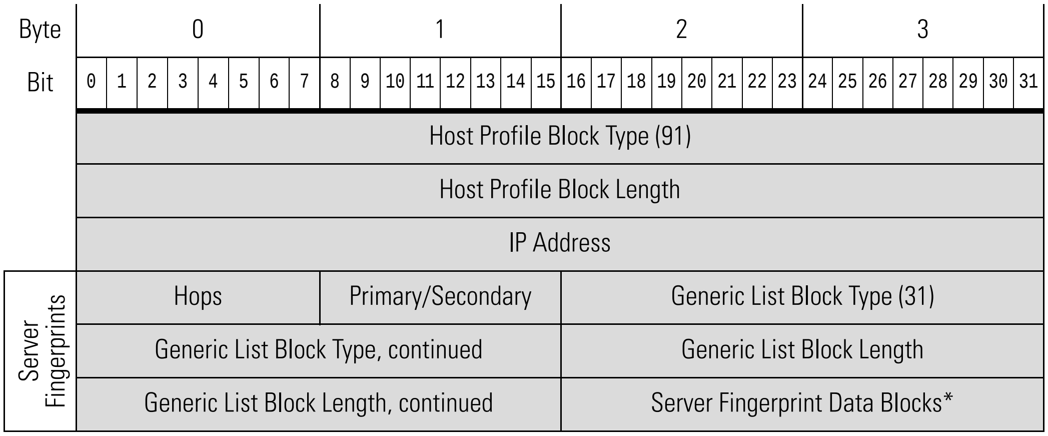
See the following sections for more information:

- Host Profile Data Block for 4.9.x - 5.0.2 on page 567

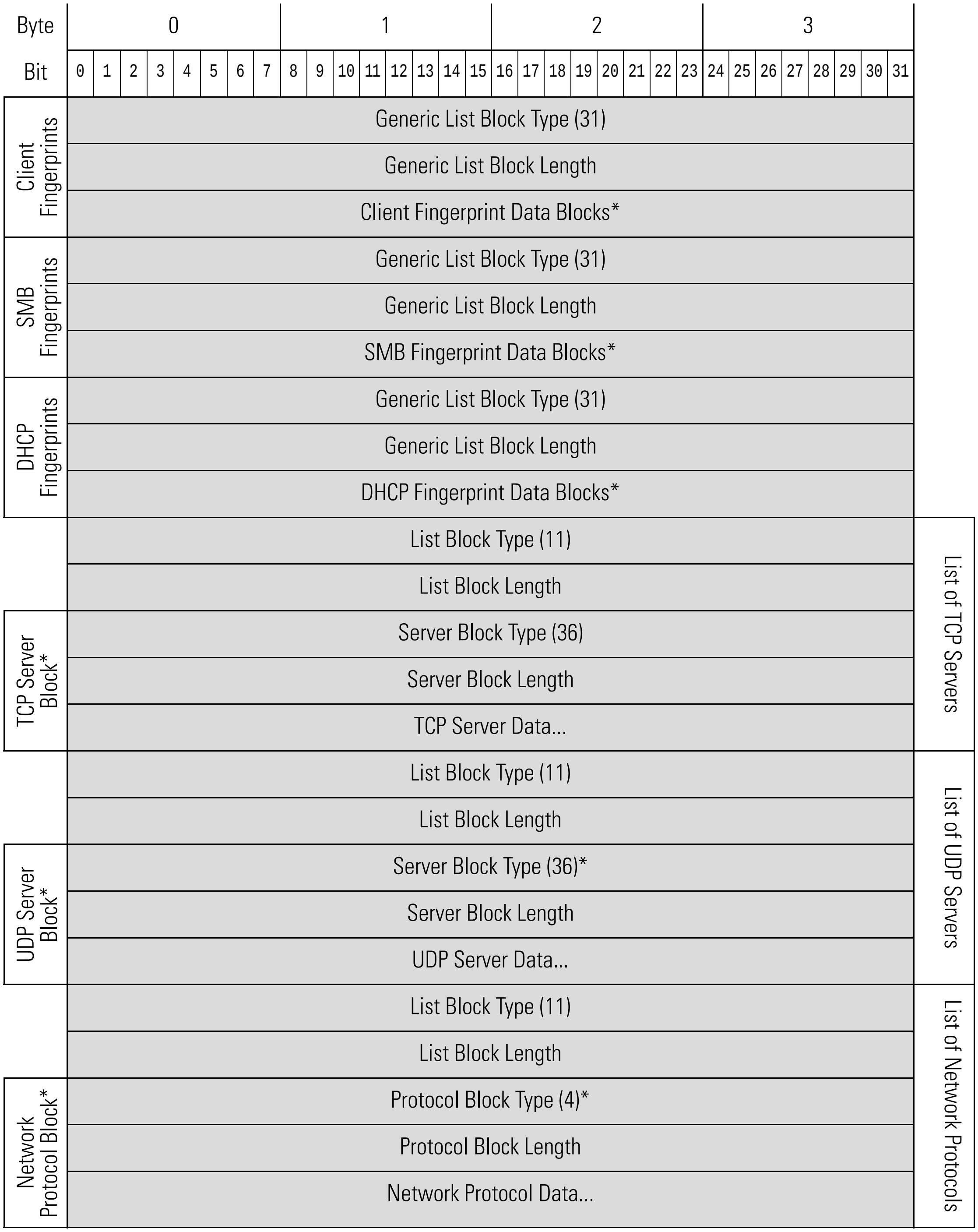
Host Profile Data Block for 4.9.x - 5.0.2

The following diagram shows the format of a Host Profile data block in 4.9 to 5.0.2. The Host Profile data block also does not include a host criticality value, but does include a VLAN presence indicator. In addition, a Host Profile data block can convey a NetBIOS name for the host. This Host Profile data block has a block type of 91.

IMPORTANT! An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Host Profile Block Type (91)			
	Host Profile Block Length			
	IP Address			
Server Fingerprints	Hops	Primary/Secondary	Generic List Block Type (31)	
	Generic List Block Type, continued		Generic List Block Length	
	Generic List Block Length, continued		Server Fingerprint Data Blocks*	



	Byte	0	1	2	3	
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
Client Fingerprints		Generic List Block Type (31)				
		Generic List Block Length				
		Client Fingerprint Data Blocks*				
SMB Fingerprints		Generic List Block Type (31)				
		Generic List Block Length				
		SMB Fingerprint Data Blocks*				
DHCP Fingerprints		Generic List Block Type (31)				
		Generic List Block Length				
		DHCP Fingerprint Data Blocks*				
		List Block Type (11)				List of TCP Servers
		List Block Length				
TCP Server Block*		Server Block Type (36)				
		Server Block Length				
		TCP Server Data...				
		List Block Type (11)				List of UDP Servers
		List Block Length				
UDP Server Block*		Server Block Type (36)*				
		Server Block Length				
		UDP Server Data...				
		List Block Type (11)				List of Network Protocols
		List Block Length				
Network Protocol Block*		Protocol Block Type (4)*				
		Protocol Block Length				
		Network Protocol Data...				

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	List Block Type (11)				List of Transport Protocols
	List Block Length				
Transport Protocol Block*	Protocol Block Type (4)*				
	Protocol Block Length				
	Transport Protocol Data...				
	List Block Type (11)				List of MAC Addresses
	List Block Length				
MAC Address Block*	MAC Address Block Type (95)*				
	MAC Address Block Length				
	MAC Address Data...				
	Host Last Seen				
	Host Type				
	VLAN Presence	VLAN ID		VLAN Type	
	VLAN Priority	Generic List Block Type (31)			List of Client Applications
	Generic List Block Type, continued	Generic List Block Length			
Client App Data	Generic List Block Length, continued	Client Application Block Type (112)*			
	Client App Block Type (29)*, con't	Client Application Block Length			
	Client Application Block Length, con't	Client Application Data...			
NetBIOS Name	String Block Type (0)				
	String Block Length				
	NetBIOS String Data...				

The Host Profile Data Block for 4.9 - 5.0.2 Fields table describes the fields of the host profile data block returned by version 4.9 to version 5.0.2.

Host Profile Data Block for 4.9 - 5.0.2 Fields

FIELD	DATA TYPE	DESCRIPTION
Host Profile Block Type	uint32	Initiates the Host Profile data block for 4.9 to 5.0.2. This data block has a block type of 91.
Host Profile Block Length	uint32	Number of bytes in the Host Profile data block, including eight bytes for the host profile block type and length fields, plus the number of bytes included in the host profile data that follows.
IP Address	uint8[4]	IP address of the host described in the profile, in IP address octets.
Hops	uint8	Number of hops from the host to the device.
Primary/ Secondary	uint8	Indicates whether the host is in the primary or secondary network of the device that detected it: • 0 — host is in the primary network. • 1 — host is in the secondary network.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Host Profile Data Block for 4.9 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an SMB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (SMB Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an SMB fingerprint. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a DHCP fingerprint. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying TCP server data. This value is always 11.

Host Profile Data Block for 4.9 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
Server Block Type	uint32	Initiates a Server data block. This value is always 89.
Server Block Length	uint32	Number of bytes in the Server data block, including eight bytes for the server block type and length fields, plus the number of bytes of TCP server data that follows.
TCP Server Data	variable	Data fields describing a TCP server, as documented in Host Server Data Block for Version 4.9.0.x on page 516.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying UDP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
Server Block Type	uint32	Initiates a Server data block describing a UDP server. This value is always 89.
Server Block Length	uint32	Number of bytes in the Server data block, including eight bytes for the server block type and length fields, plus the number of bytes of UDP server data that follows.
UDP Server Data	variable	Data fields describing a UDP server, as documented in Host Server Data Block for Version 4.9.0.x on page 516.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.

Host Profile Data Block for 4.9 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more Protocol data blocks.
Protocol Block Type	uint32	Initiates a Protocol data block describing a network protocol. This value is always 4.
Protocol Block Length	uint32	Number of bytes in the Protocol data block, including eight bytes for the protocol block type and length fields, plus the number of bytes in the protocol data that follows.
Network Protocol Data	uint16	Data field containing a network protocol number, as documented in Protocol Data Block on page 243.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more transport protocol data blocks.
Protocol Block Type	uint32	Initiates a Protocol data block describing a transport protocol. This value is always 4.
Protocol Block Length	uint32	Number of bytes in the protocol data block, including eight bytes for the protocol block type and length, plus the number of bytes in the protocol data that follows.
Transport Protocol Data	variable	Data field containing a transport protocol number, as documented in Protocol Data Block on page 243.
List Block Type	uint32	Initiates a List data block comprising MAC Address data blocks. This value is always 11.

Host Profile Data Block for 4.9 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated MAC Address data blocks.
Host MAC Address Block Type	uint32	Initiates a Host MAC Address data block. This value is always 95.
Host MAC Address Block Length	uint32	Number of bytes in the Host MAC Address data block, including eight bytes for the Host MAC address block type and length fields, plus the number of bytes in the Host MAC address data that follows.
Host MAC Address Data	variable	Host MAC address data fields described in Host MAC Address 4.9+ on page 297.
Host Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates the host type. The following values may appear: • 0 — host • 1 — router • 2 — bridge • 3 — NAT device • 4 — LB (load balancer)
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Client Application data blocks conveying client application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated client application data blocks.

Host Profile Data Block for 4.9 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Client Application Block Type	uint32	Initiates a client application block. This value is always 5.
Client Application Block Length	uint32	Number of bytes in the client application block, including eight bytes for the client application block type and length fields, plus the number of bytes in the client application data that follows.
Client Application Data	variable	Client application data fields describing a client application, as documented in Host Client Application Data Block for 5.0+ on page 334.
String Block Type	uint32	Initiates a string data block for the NetBIOS name. This value is set to 0 to indicate string data.
String Block Length	uint32	Indicates the number of bytes in the NetBIOS name data block, including eight bytes for the string block type and length, plus the number of bytes in the NetBIOS name.
NetBIOS String Data	Variable	Contains the NetBIOS name of the host described in the host profile.

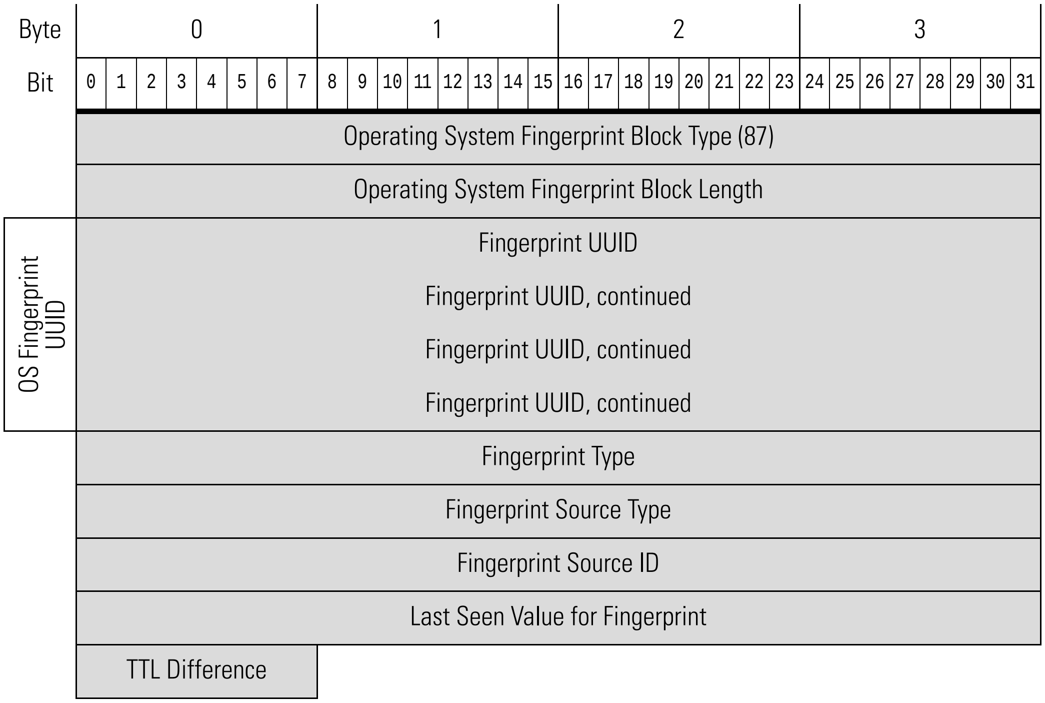
Legacy OS Fingerprint Data Blocks

See the following sections for more information:

- Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575

Operating System Fingerprint Data Block for 4.9.x - 5.0.2

The Operating System Fingerprint data block has a block type of 87. The block includes a fingerprint Universally Unique Identifier (UUID), as well as the fingerprint type, the fingerprint source type, and the fingerprint source ID. The following diagram shows the format of an Operating System Fingerprint data block for version 4.9.x to version 5.0.2.



The Operating System Fingerprint Data Block Fields table describes the fields of the operating system fingerprint data block.

Operating System Fingerprint Data Block Fields

Field	Data Type	Description
Operating System Fingerprint Data Block Type	uint32	Initiates the operating system data block. This value is always 87.
Operating System Data Block Length	uint32	Number of bytes in the Operating System Fingerprint data block. This value should always be 41: eight bytes for the data block type and length fields, sixteen bytes for the fingerprint UUID value, four bytes for the fingerprint type, four bytes for the fingerprint source type, four bytes for the fingerprint source ID, four bytes for the last seen value, and one byte for the TTL difference.

Operating System Fingerprint Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Fingerprint UUID	uint8[16]	Fingerprint identification number, in octets, that acts as a unique identifier for the operating system. The fingerprint UUID maps to the operating system name, vendor, and version in the vulnerability database (VDB).
Fingerprint Type	uint32	Indicates the type of fingerprint.
Fingerprint Source Type	uint32	Indicates the type (i.e., user or scanner) of the source that supplied the operating system fingerprint.
Fingerprint Source ID	uint32	Indicates the ID of the source that supplied the operating system fingerprint.
Last Seen	uint32	Indicates when the fingerprint was last seen in traffic.
TTL Difference	uint8	Indicates the difference between the TTL value in the fingerprint and the TTL value seen in the packet used to fingerprint the host.

Legacy Connection Data Structures

For more information, see the following sections:

- Connection Statistics Data Block for 4.7 - 4.9.0.x on page 577
- Connection Statistics Data Block 4.9.1 - 4.10.1 on page 581
- Connection Statistics Data Block 4.10.2.x on page 585
- Connection Statistics Data Block 5.0 - 5.0.2 on page 590
- Connection Statistics Data Block 5.1 on page 595
- Connection Chunk Data Block for 4.10.1 - 5.1 on page 610
- The Connection Chunk Data Block Fields table describes the components of the Connection Chunk data block: on page 611

Connection Statistics Data Block for 4.7 - 4.9.0.x

The Connection Statistics data block is used in Connection Data messages. Changes to the Connection Statistics data block between 3.5 and 4.7 include the use of a server identification number rather than a server name and the addition of a client application type identification number and a domain name string. The Connection Statistics data block for 4.7 - 4.9.0 has a block type of 56.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection data block for 4.7 - 4.9.0.x:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Connection Data Block Type (56)</td></tr>
<tr><td></td><td colspan="32">Connection Data Block Length</td></tr>
<tr><td></td><td colspan="32">Initiator IP Address</td></tr>
<tr><td></td><td colspan="32">Responder IP Address</td></tr>
<tr><td></td><td colspan="16">Initiator Port</td><td colspan="16">Responder Port</td></tr>
<tr><td></td><td colspan="32">First Packet Timestamp</td></tr>
<tr><td></td><td colspan="32">Last Packet Timestamp</td></tr>
<tr><td></td><td colspan="8">Connection Type</td><td colspan="24">Source Device IP Address</td></tr>
<tr><td></td><td colspan="8">Src Dev IP, cont.</td><td colspan="8">TCP Flags</td><td colspan="16">Packets Sent</td></tr>
<tr><td></td><td colspan="16">Packets Sent, cont.</td><td colspan="16">Packets Received</td></tr>
<tr><td></td><td colspan="16">Packets Received, cont.</td><td colspan="16">Bytes Sent</td></tr>
<tr><td></td><td colspan="16">Bytes Sent, cont.</td><td colspan="16">Bytes Received</td></tr>
<tr><td></td><td colspan="16">Bytes Received, cont.</td><td colspan="8">Protocol</td><td colspan="8">Server ID...</td></tr>
<tr><td></td><td colspan="24">Server ID, cont...</td><td colspan="8">Client App Type ID</td></tr>
<tr><td></td><td colspan="24">Client Application Type ID cont....</td><td colspan="8">Client App ID</td></tr>
<tr><td rowspan="4">Client App Version</td><td colspan="24">Client Application ID cont....</td><td colspan="8">Block Type (0)</td></tr>
<tr><td colspan="24">String Block Type (0)</td><td colspan="8">Block Length</td></tr>
<tr><td colspan="24">String Block Length</td><td colspan="8">App Version...</td></tr>
<tr><td colspan="32">Client Application Version...</td></tr>
</table>

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Client App URL	String Block Type (0)			
	String Block Length			
	Client Application URL...			
Domain Name	String Block Type (0)			
	String Block Length			
	Domain Name....			

The Connection Statistics Data Block 4.7 - 4.9.0.x Fields table describes the fields of the Connection Statistics data block returned by version 4.7.

Connection Statistics Data Block 4.7 - 4.9.0.x Fields

Field	Data Type	Description
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 4.7+. The value is always 56.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Initiator IP Address	uint8[4]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[4]	IP address of the host that responded to the initiating host, in IP address octets.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
First Packet Timestamp	uint32	UNIX timestamp that represents the date and time that the first packet was exchanged in the session.

Connection Statistics Data Block 4.7 - 4.9.0.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Last Packet Timestamp	uint32	UNIX timestamp that represents the date and time that the last packet was exchanged in the session.
Connection Type	uint8	Indicates the type of connection.
Source Device IP Address	uint8[4]	IP address of the sensor that detected the connection event, in IP address octets.
TCP Flags	uint8	Indicates any TCP flags for the connection event.
Packets Sent	uint32	Indicates the number of packets transmitted by the initiating host.
Packets Received	uint32	Number of packets transmitted by the responding host.
Bytes Sent	uint32	Number of bytes transmitted by the initiating host.
Bytes Received	uint32	Number of bytes transmitted by the responding host.
Protocol	uint8	Protocol used within the session.
Server ID	uint32	Indicates the identification number for the server.
Client Application Type ID	uint32	Identification number of the detected client application type, if applicable.
Client Application ID	uint32	Identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the client application version String data block, including eight bytes for the string block type and length fields plus the number of bytes in the client application version string.

Connection Statistics Data Block 4.7 - 4.9.0.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Client Application Version	string	Version of the client application (5.0, 5.5, and so on).
String Block Type	uint32	Initiates a string data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a string data block for the domain name. This value is always 0.
String Block Length	uint32	Number of bytes in the domain name String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the domain name string.
Domain Name	string	The domain name for the initiating host, if applicable.

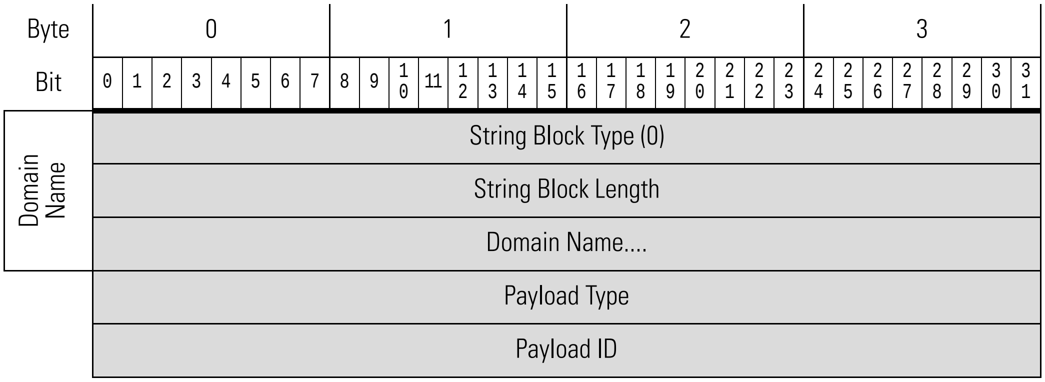
Connection Statistics Data Block 4.9.1 - 4.10.1

The Connection Statistics data block is used in Connection Data messages. Changes to the Connection data block between 4.7 and 4.9.1+ include the addition of a list of Web Application data blocks. The Connection Statistics data block for 4.9.1 - 4.10.1 has a block type of 101.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 4.9.1 - 4.10.1:

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Connection Data Block Type (101)</td></tr>
<tr><td></td><td colspan="32">Connection Data Block Length</td></tr>
<tr><td></td><td colspan="32">Initiator IP Address</td></tr>
<tr><td></td><td colspan="32">Responder IP Address</td></tr>
<tr><td></td><td colspan="16">Initiator Port</td><td colspan="16">Responder Port</td></tr>
<tr><td></td><td colspan="32">First Packet Timestamp</td></tr>
<tr><td></td><td colspan="32">Last Packet Timestamp</td></tr>
<tr><td></td><td colspan="8">Connection Type</td><td colspan="24">Source Device IP Address</td></tr>
<tr><td></td><td colspan="8">Src Dev IP, cont.</td><td colspan="8">TCP Flags</td><td colspan="16">Packets Sent</td></tr>
<tr><td></td><td colspan="16">Packets Sent, cont.</td><td colspan="16">Packets Received</td></tr>
<tr><td></td><td colspan="16">Packets Received, cont.</td><td colspan="16">Bytes Sent</td></tr>
<tr><td></td><td colspan="16">Bytes Sent, cont.</td><td colspan="16">Bytes Received</td></tr>
<tr><td></td><td colspan="16">Bytes Received, cont.</td><td colspan="8">Protocol</td><td colspan="8">Server ID...</td></tr>
<tr><td></td><td colspan="24">Server ID, cont...</td><td colspan="8">Client App Type ID</td></tr>
<tr><td></td><td colspan="24">Client Application Type ID cont....</td><td colspan="8">Client App ID</td></tr>
<tr><td rowspan="4">Client App Version</td><td colspan="24">Client Application ID cont....</td><td colspan="8">Block Type (0)</td></tr>
<tr><td colspan="24">String Block Type (0)</td><td colspan="8">Block Length</td></tr>
<tr><td colspan="24">String Block Length</td><td colspan="8">App Version...</td></tr>
<tr><td colspan="32">Client Application Version...</td></tr>
<tr><td rowspan="3">Client App URL</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">Client Application URL...</td></tr>
</table>



The Connection Statistics Data Block 4.9.1 - 4.10.1 Fields table describes the fields of the Connection Statistics data block returned by 4.9.1 - 4.10.x1

Connection Statistics Data Block 4.9.1 - 4.10.1 Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 4.9.1+. The value is always 101.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Initiator IP Address	uint8[4]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[4]	IP address of the host that responded to the initiating host, in IP address octets.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
First Packet Timestamp	uint32	UNIX timestamp that represents the date and time that the first packet was exchanged in the session.

Connection Statistics Data Block 4.9.1 - 4.10.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Last Packet Timestamp	uint32	UNIX timestamp that represents the date and time that the last packet was exchanged in the session.
Connection Type	uint8	Indicates the type of connection.
Source Device IP Address	uint8[4]	IP address of the sensor that detected the connection event, in IP address octets.
TCP Flags	uint8	Indicates any TCP flags for the connection event.
Packets Sent	uint32	Indicates the number of packets transmitted by the initiating host.
Packets Received	uint32	Number of packets transmitted by the responding host.
Bytes Sent	uint32	Number of bytes transmitted by the initiating host.
Bytes Received	uint32	Number of bytes transmitted by the responding host.
Protocol	uint8	Protocol used within the session.
Server ID	uint32	Indicates the identification number for the server.
Client Application Type ID	uint32	Identification number of the detected client application type, if applicable.
Client Application ID	uint32	Identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the client application version String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application version string.

Connection Statistics Data Block 4.9.1 - 4.10.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Client Application Version	string	Version of the client application (5.0, 5.5, and so on).
String Block Type	uint32	Initiates a string data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a string data block for the domain name. This value is always 0.
String Block Length	uint32	Number of bytes in the domain name String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the domain name string.
Domain Name	string	The domain name for the initiating host, if applicable.
Payload Type	uint32	Indicates the type of the payload data.
Payload ID	uint32	Indicates the ID of the payload.

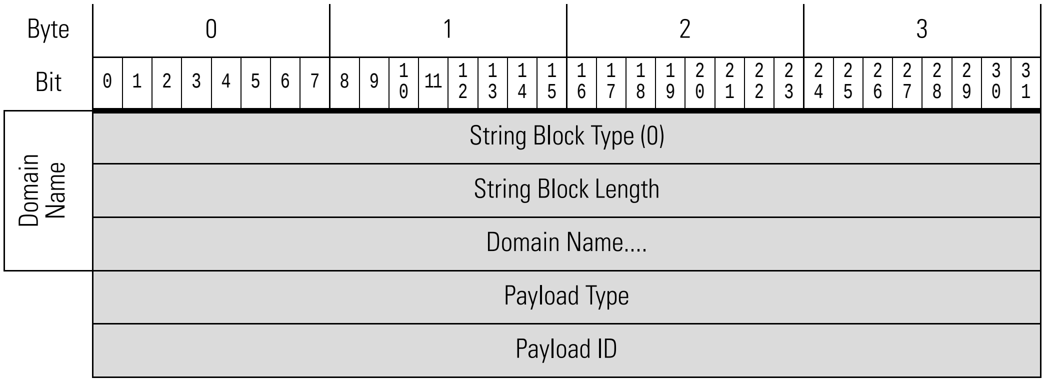
Connection Statistics Data Block 4.10.2.x

The Connection Statistics data block is used in Connection Data messages. Changes to the Connection data block between 4.10.1 and 4.10.2 include the addition of NetFlow fields. The Connection Statistics data block for 4.10.2.x has a block type of 125.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 4.10.2.x:

	Byte 0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Connection Data Block Type (125)			
	Connection Data Block Length			
	Initiator IP Address			
	Responder IP Address			
	Initiator Port		Responder Port	
	First Packet Timestamp			
	Last Packet Timestamp			
	Connection Type	NetFlow Src TOS	NetFlow Dst TOS	NetFlow SNMP Input
	NetFlow SNMP Input cont.	NetFlow SNMP Output		Source Device IP Address
	Source Device IP Address cont.			TCP Flags
	Packets Sent			
	Packets Received			
	Bytes Sent			
	Bytes Received			
	Protocol	Server ID...		
	Server ID, cont...	Client App Type ID		
	Client Application Type ID cont....	Client App ID		
Client App Version	Client Application ID cont....	String Block Type (0)		
	Block Type cont.	String Block Length		
	String Block Length	Client Application Version...		
Client App URL	String Block Type (0)			
	String Block Length			
	Client Application URL...			



The Connection Statistics Data Block 4.10.2 Fields table describes the fields of the Connection Statistics data block returned by 4.10.2.

Connection Statistics Data Block 4.10.2 Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 4.10.2.x. The value is always 125.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Initiator IP Address	uint8[4]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[4]	IP address of the host that responded to the initiating host, in IP address octets.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
First Packet Timestamp	uint32	UNIX timestamp that represents the date and time that the first packet was exchanged in the session.

Connection Statistics Data Block 4.10.2 Fields (Continued)

Field	Data Type	Description
Last Packet Timestamp	uint32	UNIX timestamp that represents the date and time that the last packet was exchanged in the session.
Connection Type	uint8	Indicates the type of connection.
NetFlow Source TOS	uint8	Type of service from the IP header when packets are flowing from the source to the destination.
NetFlow Destination TOS	uint8	Type of service from the IP header when packets are flowing from the destination to the source
NetFlow SNMP Input	uint16	ID of the interface used by packets flowing from the source to the destination.
NetFlow SNMP Output	uint16	ID of the interface used by packets flowing from the destination to the source.
Source Device IP Address	uint8[4]	IP address of the sensor that detected the connection event, in IP address octets.
TCP Flags	uint8	Indicates any TCP flags for the connection event.
Packets Sent	uint32	Indicates the number of packets transmitted by the initiating host.
Packets Received	uint32	Number of packets transmitted by the responding host.
Bytes Sent	uint32	Number of bytes transmitted by the initiating host.
Bytes Received	uint32	Number of bytes transmitted by the responding host.
Protocol	uint8	Protocol used within the session.
Server ID	uint32	Indicates the identification number for the server.

Connection Statistics Data Block 4.10.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Client Application Type ID	uint32	Identification number of the detected client application type, if applicable.
Client Application ID	uint32	Identification number of the detected client application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the client application version String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application version string.
Client Application Version	string	Version of the client application (5.0, 5.5, and so on).
String Block Type	uint32	Initiates a string data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a string data block for the domain name. This value is always 0.
String Block Length	uint32	Number of bytes in the domain name String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the domain name string.
Domain Name	string	The domain name for the initiating host, if applicable.

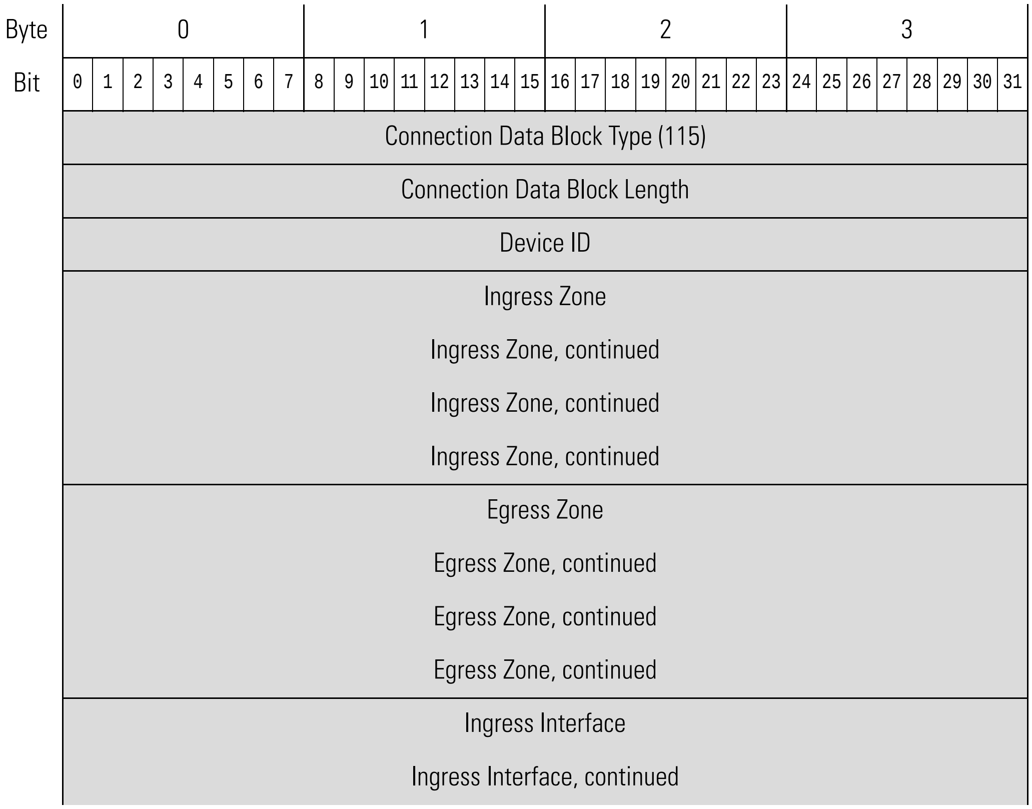
Connection Statistics Data Block 4.10.2 Fields (Continued)

Field	Data Type	Description
Payload Type	uint32	Indicates the type of the payload data.
Payload ID	uint32	Indicates the ID of the payload.

Connection Statistics Data Block 5.0 - 5.0.2

The Connection Statistics data block is used in Connection Data messages. Changes to the Connection data block between 4.10.x and 5.0 include addition of new fields with configuration parameters introduced in 5.0 (security zone, ingress and egress interface, URL category and reputation, and user, plus fields for additional tracking information such as violated policy and rule). The Connection Statistics data block for version 5.0 - 5.0.2 has a block type of 115. For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 5.0 - 5.0.2:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Ingress Interface, continued			
	Ingress Interface, continued			
	Egress Interface			
	Egress Interface, continued			
	Egress Interface, continued			
	Egress Interface, continued			
	Initiator IP Address			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Responder IP Address			
	Responder IP Address, continued			
	Responder IP Address, continued			
	Responder IP Address, continued			
	Policy Revision			
	Policy Revision, continued			
	Policy Revision, continued			
	Policy Revision, continued			
	Rule ID			
	Rule Action			
	Initiator Port		Responder Port	
	TCP Flags		Protocol	NetFlow Source
	NetFlow Source, continued			
	NetFlow Source, continued			
	NetFlow Source, continued			
	NetFlow Source, continued			First Pkt Time

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		First Packet Timestamp, continued			Last Pkt Time
		Last Packet Timestamp, continued			Packets Sent
		Packets Sent, continued			
		Packets Sent, continued			Packets Rcvd
		Packets Received, continued			
		Packets Received, continued			Bytes Sent
		Bytes Sent, continued			
		Packets Received, continued			Bytes Rcvd
		Bytes Received, continued			
		Bytes Received, continued			User ID
		User ID, continued			Application Protocol ID
		Application Protocol ID, continued			URL Category
		URL Category, continued			URL Reputation
		URL Reputation, continued			Client App ID
		Client Application ID, continued			Web App ID
		Web Application ID, continued			String Block Type (0)
Client App URL		String Block Type, continued			String Block Length
		String Block Length, continued			Client Application URL...
NetBIOS Name		String Block Type (0)			
		String Block Length			
		NetBIOS Name....			
Client App Version		String Block Type (0)			
		String Block Length			
		Client Application Version...			

The Connection Statistics Data Block 5.0 - 5.0.2 Fields table describes the fields of the Connection Statistics data block for 5.0 - 5.0.2.

Connection Statistics Data Block 5.0 - 5.0.2 Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.0 to 5.0.2. The value is always 115.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint32	The action selected in the user interface for that rule (allow, block, and so forth).
Initiator Port	uint16	Port used by the initiating host.

Connection Statistics Data Block 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Packets Sent	uint64	Number of packets transmitted by the initiating host.
Packets Received	uint64	Number of packets transmitted by the responding host.
Bytes Sent	uint64	Number of bytes transmitted by the initiating host.
Bytes Received	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.

Connection Statistics Data Block 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.

Connection Statistics Data Block 5.1

The Connection Statistics data block is used in Connection Data messages. Changes to the Connection data block between 5.0.2 and 5.1 include the addition of new fields with configuration parameters introduced in 5.1 (rule action reason, monitor rules, Security Intelligence source/destination, Security Intelligence layer). The Connection Statistics data block for version 5.1 has a block type of 126.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 5.1:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Connection Data Block Type (126)			
	Connection Data Block Length			
	Device ID			
	Ingress Zone			
	Ingress Zone, continued			
	Ingress Zone, continued			
	Ingress Zone, continued			
	Egress Zone			
	Egress Zone, continued			
	Egress Zone, continued			
	Egress Zone, continued			
	Ingress Interface			
	Ingress Interface, continued			
	Ingress Interface, continued			
	Ingress Interface, continued			
	Egress Interface			
	Egress Interface, continued			
	Egress Interface, continued			
	Egress Interface, continued			
	Initiator IP Address			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Initiator IP Address, continued			

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Responder IP Address</td></tr>
<tr><td></td><td colspan="32">Responder IP Address, continued</td></tr>
<tr><td></td><td colspan="32">Responder IP Address, continued</td></tr>
<tr><td></td><td colspan="32">Responder IP Address, continued</td></tr>
<tr><td></td><td colspan="32">Policy Revision</td></tr>
<tr><td></td><td colspan="32">Policy Revision, continued</td></tr>
<tr><td></td><td colspan="32">Policy Revision, continued</td></tr>
<tr><td></td><td colspan="32">Policy Revision, continued</td></tr>
<tr><td></td><td colspan="32">Rule ID</td></tr>
<tr><td></td><td colspan="16">Rule Action</td><td colspan="16">Rule Reason</td></tr>
<tr><td></td><td colspan="16">Initiator Port</td><td colspan="16">Responder Port</td></tr>
<tr><td></td><td colspan="16">TCP Flags</td><td colspan="8">Protocol</td><td colspan="8">NetFlow Source</td></tr>
<tr><td></td><td colspan="32">NetFlow Source, continued</td></tr>
<tr><td></td><td colspan="32">NetFlow Source, continued</td></tr>
<tr><td></td><td colspan="32">NetFlow Source, continued</td></tr>
<tr><td></td><td colspan="24">NetFlow Source, continued</td><td colspan="8">First Pkt Time</td></tr>
<tr><td></td><td colspan="24">First Packet Timestamp, continued</td><td colspan="8">Last Pkt Time</td></tr>
<tr><td></td><td colspan="24">Last Packet Timestamp, continued</td><td colspan="8">Initiator Transmitted Packets</td></tr>
<tr><td></td><td colspan="32">Initiator Transmitted Packets, continued</td></tr>
<tr><td></td><td colspan="24">Initiator Transmitted Packets, continued</td><td colspan="8">Responder Transmitted Packets</td></tr>
<tr><td></td><td colspan="32">Responder Transmitted Packets, continued</td></tr>
<tr><td></td><td colspan="24">Responder Transmitted Packets, continued</td><td colspan="8">Initiator Transmitted Bytes</td></tr>
<tr><td></td><td colspan="32">Initiator Transmitted Bytes, continued</td></tr>
<tr><td></td><td colspan="24">Initiator Transmitted Bytes, continued</td><td colspan="8">Responder Transmitted Bytes</td></tr>
</table>

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Responder Transmitted Bytes, continued</td></tr>
<tr><td></td><td colspan="24">Responder Transmitted Bytes, continued</td><td colspan="8">User ID</td></tr>
<tr><td></td><td colspan="24">User ID, continued</td><td colspan="8">Application Protocol ID</td></tr>
<tr><td></td><td colspan="24">Application Protocol ID, continued</td><td colspan="8">URL Category</td></tr>
<tr><td></td><td colspan="24">URL Category, continued</td><td colspan="8">URL Reputation</td></tr>
<tr><td></td><td colspan="24">URL Reputation, continued</td><td colspan="8">Client App ID</td></tr>
<tr><td></td><td colspan="24">Client Application ID, continued</td><td colspan="8">Web App ID</td></tr>
<tr><td></td><td colspan="24">Web Application ID, continued</td><td colspan="8">String Block Type (0)</td></tr>
<tr><td rowspan="2">Client App URL</td><td colspan="24">String Block Type, continued</td><td colspan="8">String Block Length</td></tr>
<tr><td colspan="24">String Block Length, continued</td><td colspan="8">Client Application URL...</td></tr>
<tr><td rowspan="3">NetBIOS Name</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">NetBIOS Name....</td></tr>
<tr><td rowspan="3">Client App Version</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">Client Application Version...</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 1</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 2</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 3</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 4</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 5</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 6</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 7</td></tr>
<tr><td></td><td colspan="32">Monitor Rule 8</td></tr>
<tr><td></td><td colspan="8">Sec. Int. Src/Dst</td><td colspan="8">Sec. Int. Rep Layer</td><td colspan="16"></td></tr>
</table>

The Connection Statistics Data Block 5.1 Fields table describes the fields of the Connection Statistics data block for 5.1.

Connection Statistics Data Block 5.1 Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.1+. The value is always 126.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.

Connection Statistics Data Block 5.1 Fields (Continued)

Field	Data Type	Description
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.

Connection Statistics Data Block 5.1 Fields (Continued)

Field	Data Type	Description
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.

Connection Statistics Data Block 5.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.

Connection Statistics Data Block 5.2.x

The connection statistics data block is used in connection data messages. Changes to the connection data block between versions 5.1.1 and 5.2 include the addition of new fields to support geolocation. The connection statistics data block for version 5.2.x has a block type of 144 in the series 1 group of blocks. It deprecates block type 137, The Connection Chunk Data Block Fields table describes the components of the Connection Chunk data block: on page 611.

For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 5.2+:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Connection Data Block Type (144)			
	Connection Data Block Length			
	Device ID			
	Ingress Zone			
	Ingress Zone, continued			
	Ingress Zone, continued			
	Ingress Zone, continued			
	Egress Zone			
	Egress Zone, continued			
	Egress Zone, continued			
	Egress Zone, continued			
	Ingress Interface			
	Ingress Interface, continued			
	Ingress Interface, continued			
	Ingress Interface, continued			
	Egress Interface			
	Egress Interface, continued			
	Egress Interface, continued			
	Egress Interface, continued			
	Initiator IP Address			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Responder IP Address			
	Responder IP Address, continued			
	Responder IP Address, continued			

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="32">Responder IP Address, continued</td></tr>
<tr><td></td><td colspan="32">Policy Revision</td></tr>
<tr><td></td><td colspan="32">Policy Revision, continued</td></tr>
<tr><td></td><td colspan="32">Policy Revision, continued</td></tr>
<tr><td></td><td colspan="32">Policy Revision, continued</td></tr>
<tr><td></td><td colspan="32">Rule ID</td></tr>
<tr><td></td><td colspan="16">Rule Action</td><td colspan="16">Rule Reason</td></tr>
<tr><td></td><td colspan="16">Initiator Port</td><td colspan="16">Responder Port</td></tr>
<tr><td></td><td colspan="16">TCP Flags</td><td colspan="8">Protocol</td><td colspan="8">NetFlow Source</td></tr>
<tr><td></td><td colspan="32">NetFlow Source, continued</td></tr>
<tr><td></td><td colspan="32">NetFlow Source, continued</td></tr>
<tr><td></td><td colspan="32">NetFlow Source, continued</td></tr>
<tr><td></td><td colspan="24">NetFlow Source, continued</td><td colspan="8">Instance ID</td></tr>
<tr><td></td><td colspan="8">Instance ID, cont.</td><td colspan="16">Connection Counter</td><td colspan="8">First Pkt Time</td></tr>
<tr><td></td><td colspan="24">First Packet Timestamp, continued</td><td colspan="8">Last Pkt Time</td></tr>
<tr><td></td><td colspan="24">Last Packet Timestamp, continued</td><td colspan="8">Initiator Tx Packets</td></tr>
<tr><td></td><td colspan="32">Initiator Transmitted Packets, continued</td></tr>
<tr><td></td><td colspan="24">Initiator Transmitted Packets, continued</td><td colspan="8">Resp. Tx Packets</td></tr>
<tr><td></td><td colspan="32">Responder Transmitted Packets, continued</td></tr>
<tr><td></td><td colspan="24">Responder Transmitted Packets, continued</td><td colspan="8">Initiator Tx Bytes</td></tr>
<tr><td></td><td colspan="32">Initiator Transmitted Bytes, continued</td></tr>
<tr><td></td><td colspan="24">Initiator Transmitted Bytes, continued</td><td colspan="8">Resp. Tx Bytes</td></tr>
<tr><td></td><td colspan="32">Responder Transmitted Bytes, continued</td></tr>
<tr><td></td><td colspan="24">Responder Transmitted Bytes, continued</td><td colspan="8">User ID</td></tr>
<tr><td></td><td colspan="24">User ID, continued</td><td colspan="8">Application Prot. ID</td></tr>
<tr><td></td><td colspan="24">Application Protocol ID, continued</td><td colspan="8">URL Category</td></tr>
</table>

<table>
<tr><td></td><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td></td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td></td><td colspan="24">URL Category, continued</td><td colspan="8">URL Reputation</td></tr>
<tr><td></td><td></td><td colspan="24">URL Reputation, continued</td><td colspan="8">Client App ID</td></tr>
<tr><td></td><td></td><td colspan="24">Client Application ID, continued</td><td colspan="8">Web App ID</td></tr>
<tr><td rowspan="3">Client URL</td><td></td><td colspan="24">Web Application ID, continued</td><td colspan="8">Str. Block Type (0)</td></tr>
<tr><td></td><td colspan="24">String Block Type, continued</td><td colspan="8">String Block Length</td></tr>
<tr><td></td><td colspan="24">String Block Length, continued</td><td colspan="8">Client App. URL...</td></tr>
<tr><td rowspan="3">NetBIOS Name</td><td></td><td colspan="32">String Block Type (0)</td></tr>
<tr><td></td><td colspan="32">String Block Length</td></tr>
<tr><td></td><td colspan="32">NetBIOS Name...</td></tr>
<tr><td rowspan="3">Client App Version</td><td></td><td colspan="32">String Block Type (0)</td></tr>
<tr><td></td><td colspan="32">String Block Length</td></tr>
<tr><td></td><td colspan="32">Client Application Version...</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 1</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 2</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 3</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 4</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 5</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 6</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 7</td></tr>
<tr><td></td><td></td><td colspan="32">Monitor Rule 8</td></tr>
<tr><td></td><td></td><td colspan="8">Sec. Int. Src/Dst</td><td colspan="8">Sec. Int. Layer</td><td colspan="16">File Event Count</td></tr>
<tr><td></td><td></td><td colspan="16">Intrusion Event Count</td><td colspan="16">Initiator Country</td></tr>
<tr><td></td><td></td><td colspan="16">Responder Country</td><td colspan="16"></td></tr>
</table>

The Connection Statistics Data Block 5.2.x Fields table describes the fields of the Connection Statistics data block for 5.2.x:

Connection Statistics Data Block 5.2.x Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.2+. The value is always 144.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.

Connection Statistics Data Block 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.

Connection Statistics Data Block 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.

Connection Statistics Data Block 5.2.x Fields (Continued)

Field	Data Type	Description
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.
Initiator Country	uint16	Code for the country of the initiating host.
Responder Country	uint16	Code for the country of the responding host.

Connection Chunk Data Block for 4.10.1 - 5.1

The Connection Chunk data block conveys connection data detected by a NetFlow device. The Connection Chunk data block has a block type of 66 for pre-4.10.1 versions. For version 4.10.1 - 5.1, it has a block type of 119.

The following diagram shows the format of the Connection Chunk data block:

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Connection Chunk Block Type (66 | 119)			
	Connection Chunk Block Length			
	Initiator IP Address			
	Responder IP Address			
	Start Time			
	Application ID			
	Responder Port		Protocol	Connection Type
	NetFlow Detector IP Address			
	Packets Sent			
	Packets Received			
	Bytes Sent			
	Bytes Received			
	Connections			

The Connection Chunk Data Block Fields table describes the components of the Connection Chunk data block:

Connection Chunk Data Block Fields

Field	Data Type	Description
Connection Chunk Block Type	uint32	Initiates a Connection Chunk data block. This value is 66 for versions before 4.10.1 and a value of 119 for version 5.0+.
Connection Chunk Block Length	uint32	Total number of bytes in the Connection Chunk data block, including eight bytes for the connection chunk block type and length fields, plus the number of bytes in the connection chunk data that follows.
Initiator IP Address	uint8[4]	IP address of the host that initiated the connection, in IP address octets.
Responder IP Address	uint8[4]	IP address of the host responding in the connection, in IP address octets.
Start Time	uint32	The starting time for the connection chunk.
Application ID	uint32	Application identification number for the application protocol used in the connection.
Responder Port	uint16	The port used by the responder in the connection chunk.
Protocol	uint8	The protocol for the packet containing the user information.
Connection Type	uint8	The type of connection.
Source Device IP Address	uint8[4]	IP address of the NetFlow device that detected the connection, in IP address octets.
Packets Sent	uint32	The number of packets sent in the connection chunk.
Packets Received	uint32	The number of packets received in the connection chunk.
Bytes Sent	uint32	The number of bytes sent in the connection chunk.

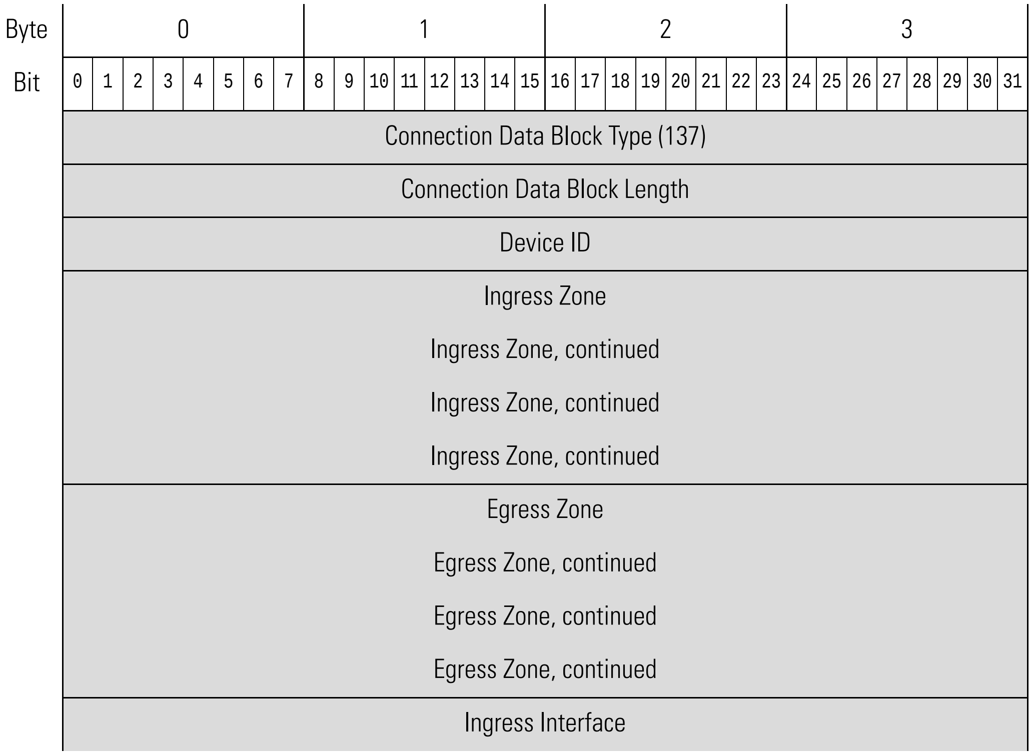
Connection Chunk Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Bytes Received	uint32	The number of bytes received in the connection chunk.
Connections	uint32	The number of connections made in the connection chunk.

Connection Statistics Data Block 5.1.1.x

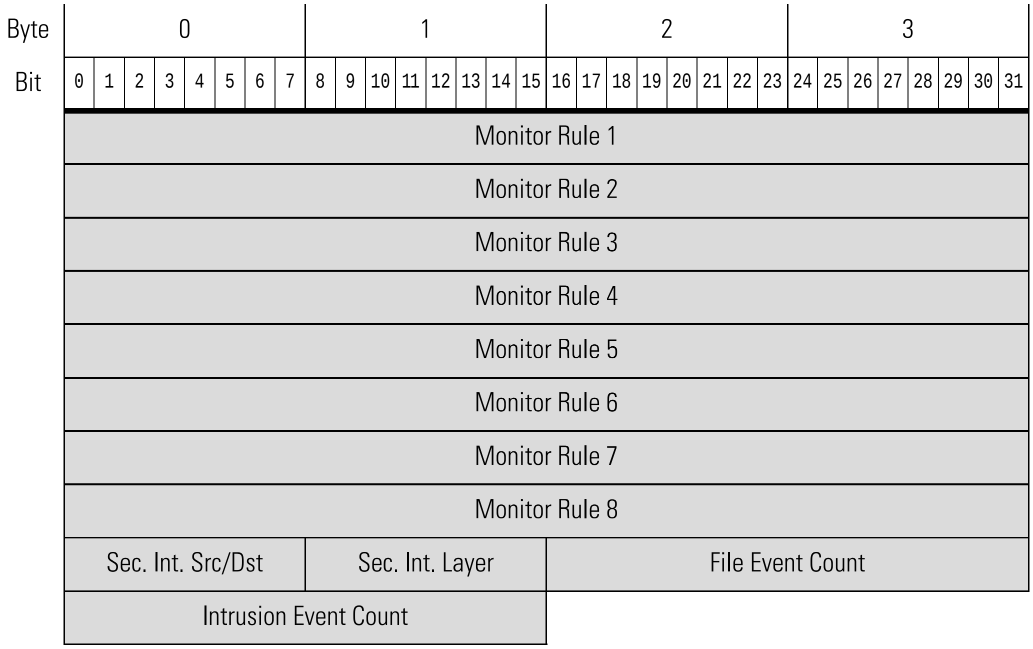
The connection statistics data block is used in connection data messages. Changes to the connection data block between versions 5.1 and 5.1.1 include the addition of new fields to identify associated intrusion events. The connection statistics data block for version 5.1.1.x has a block type of 137. It deprecates block type 126, Connection Statistics Data Block 5.1 on page 595. For more information on the Connection Statistics Data message, see Connection Statistics Data Message on page 215.

The following diagram shows the format of a Connection Statistics data block for 5.1.1:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Ingress Interface, continued			
	Ingress Interface, continued			
	Ingress Interface, continued			
	Egress Interface			
	Egress Interface, continued			
	Egress Interface, continued			
	Egress Interface, continued			
	Initiator IP Address			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Initiator IP Address, continued			
	Responder IP Address			
	Responder IP Address, continued			
	Responder IP Address, continued			
	Responder IP Address, continued			
	Policy Revision			
	Policy Revision, continued			
	Policy Revision, continued			
	Policy Revision, continued			
	Rule ID			
	Rule Action		Rule Reason	
	Initiator Port		Responder Port	
	TCP Flags		Protocol	NetFlow Source
	NetFlow Source, continued			
	NetFlow Source, continued			
	NetFlow Source, continued			

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		NetFlow Source, continued			Instance ID
		Instance ID, cont.	Connection Counter		First Pkt Time
		First Packet Timestamp, continued			Last Pkt Time
		Last Packet Timestamp, continued			Initiator Tx Packets
		Initiator Transmitted Packets, continued			
		Initiator Transmitted Packets, continued			Resp. Tx Packets
		Responder Transmitted Packets, continued			
		Responder Transmitted Packets, continued			Initiator Tx Bytes
		Initiator Transmitted Bytes, continued			
		Initiator Transmitted Bytes, continued			Resp. Tx Bytes
		Responder Transmitted Bytes, continued			
		Responder Transmitted Bytes, continued			User ID
		User ID, continued			Application Prot. ID
		Application Protocol ID, continued			URL Category
		URL Category, continued			URL Reputation
		URL Reputation, continued			Client App ID
		Client Application ID, continued			Web App ID
Client URL		Web Application ID, continued			Str. Block Type (0)
		String Block Type, continued			String Block Length
		String Block Length, continued			Client App. URL...
NetBIOS Name		String Block Type (0)			
		String Block Length			
		NetBIOS Name...			
Client App Version		String Block Type (0)			
		String Block Length			
		Client Application Version...			



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Monitor Rule 1			
	Monitor Rule 2			
	Monitor Rule 3			
	Monitor Rule 4			
	Monitor Rule 5			
	Monitor Rule 6			
	Monitor Rule 7			
	Monitor Rule 8			
	Sec. Int. Src/Dst	Sec. Int. Layer	File Event Count	
	Intrusion Event Count			

The Connection Statistics Data Block 5.1.1.x Fields table describes the fields of the Connection Statistics data block for 5.1.1.x.

Connection Statistics Data Block 5.1.1.x Fields

FIELD	DATA TYPE	DESCRIPTION
Connection Statistics Data Block Type	uint32	Initiates a Connection Statistics data block for 5.1.1.x. The value is always 137.
Connection Statistics Data Block Length	uint32	Number of bytes in the Connection Statistics data block, including eight bytes for the connection statistics block type and length fields, plus the number of bytes in the connection data that follows.
Device ID	uint32	The device that detected the connection event.
Ingress Zone	uint8[16]	Ingress security zone in the event that triggered the policy violation.
Egress Zone	uint8[16]	Egress security zone in the event that triggered the policy violation.
Ingress Interface	uint8[16]	Interface for the inbound traffic.

Connection Statistics Data Block 5.1.1.x Fields (Continued)

Field	Data Type	Description
Egress Interface	uint8[16]	Interface for the outbound traffic.
Initiator IP Address	uint8[16]	IP address of the host that initiated the session described in the connection event, in IP address octets.
Responder IP Address	uint8[16]	IP address of the host that responded to the initiating host, in IP address octets.
Policy Revision	uint8[16]	Revision number of the rule associated with the triggered correlation event, if applicable.
Rule ID	uint32	Internal identifier for the rule that triggered the event, if applicable.
Rule Action	uint16	The action selected in the user interface for that rule (allow, block, and so forth).
Rule Reason	uint16	The reason the rule triggered the event.
Initiator Port	uint16	Port used by the initiating host.
Responder Port	uint16	Port used by the responding host.
TCP Flags	uint16	Indicates any TCP flags for the connection event.
Protocol	uint8	The IANA-specified protocol number.
NetFlow Source	uint8[16]	IP address of the NetFlow-enabled device that exported the data for the connection.
Instance ID	uint16	Numerical ID of the Snort instance on the managed device that generated the event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
First Packet Timestamp	uint32	UNIX timestamp of the date and time the first packet was exchanged in the session.
Last Packet Timestamp	uint32	UNIX timestamp of the date and time the last packet was exchanged in the session.

Connection Statistics Data Block 5.1.1.x Fields (Continued)

Field	Data Type	Description
Initiator Transmitted Packets	uint64	Number of packets transmitted by the initiating host.
Responder Transmitted Packets	uint64	Number of packets transmitted by the responding host.
Initiator Transmitted Bytes	uint64	Number of bytes transmitted by the initiating host.
Responder Transmitted Bytes	uint64	Number of bytes transmitted by the responding host.
User ID	uint32	Internal identification number for the user who last logged into the host that generated the traffic.
Application Protocol ID	uint32	Application ID of the application protocol.
URL Category	uint32	The internal identification number of the URL category.
URL Reputation	uint32	The internal identification number for the URL reputation.
Client Application ID	uint32	The internal identification number of the detected client application, if applicable.
Web Application ID	uint32	The internal identification number of the detected web application, if applicable.
String Block Type	uint32	Initiates a String data block for the client application URL. This value is always 0.
String Block Length	uint32	Number of bytes in the client application URL String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the client application URL string.
Client Application URL	string	URL the client application accessed, if applicable (/files/index.html, for example).

Connection Statistics Data Block 5.1.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for the client application version. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block for the client application version, including eight bytes for the string block type and length, plus the number of bytes in the version.
Client Application Version	string	Client application version.
Monitor Rule 1	uint32	The ID of the first monitor rule associated with the connection event.
Monitor Rule 2	uint32	The ID of the second monitor rule associated with the connection event.
Monitor Rule 3	uint32	The ID of the third monitor rule associated with the connection event.
Monitor Rule 4	uint32	The ID of the fourth monitor rule associated with the connection event.
Monitor Rule 5	uint32	The ID of the fifth monitor rule associated with the connection event.
Monitor Rule 6	uint32	The ID of the sixth monitor rule associated with the connection event.
Monitor Rule 7	uint32	The ID of the seventh monitor rule associated with the connection event.
Monitor Rule 8	uint32	The ID of the eighth monitor rule associated with the connection event.

Connection Statistics Data Block 5.1.1.x Fields (Continued)

Field	Data Type	Description
Security Intelligence Source/ Destination	uint8	Whether the source or destination IP address matched the IP blacklist.
Security Intelligence Layer	uint8	The IP layer that matched the IP blacklist.
File Event Count	uint16	Value used to distinguish between file events that happen during the same second.
Intrusion Event Count	uint16	Value used to distinguish between intrusion events that happen during the same second.

Legacy File Event Data Structures

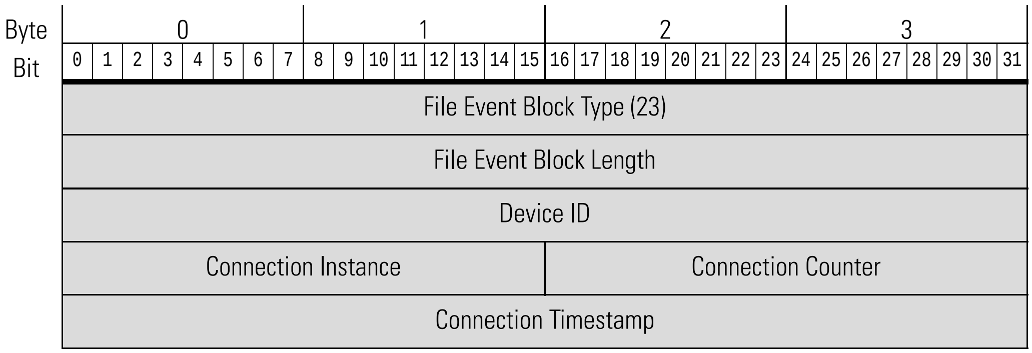
The following topics describe other legacy file event data structures:

- File Event for 5.1.1.x on page 619
- File Event for 5.2.x on page 623
- File Event SHA Hash for 5.1.1-5.2.x on page 628

File Event for 5.1.1.x

The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 23 in the series 2 group of blocks.

The following graphic shows the structure of the File Event data block.:



<table>
<tr><td></td><td colspan="4">File Event Timestamp</td></tr>
<tr><td></td><td colspan="4">Source IP Address</td></tr>
<tr><td></td><td colspan="4">Source IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Source IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Source IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Destination IP Address</td></tr>
<tr><td></td><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td></td><td colspan="4">Destination IP Address, continued</td></tr>
<tr><td></td><td>Disposition</td><td>Action</td><td colspan="2">SHA Hash</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="4">SHA Hash, continued</td></tr>
<tr><td></td><td colspan="2">SHA Hash, continued</td><td colspan="2">File Type ID</td></tr>
<tr><td rowspan="3">File Name</td><td colspan="2">File Type ID, cont.</td><td colspan="2">String Block Type (0)</td></tr>
<tr><td colspan="2">String Block Type (0), cont.</td><td colspan="2">String Block Length</td></tr>
<tr><td colspan="2">String Block Length, cont.</td><td colspan="2">File Name...</td></tr>
<tr><td></td><td colspan="4">File Size</td></tr>
<tr><td></td><td colspan="4">File Size, continued</td></tr>
<tr><td></td><td>Direction</td><td colspan="3">Application ID</td></tr>
<tr><td></td><td>App ID, cont.</td><td colspan="3">User ID</td></tr>
</table>

URI	User ID, cont.	String Block Type (0)		
	String Block Type (0), cont.	String Block Length		
	String Block Length, cont.	URI...		
Signature	String Block Type (0)			
	String Block Length			
	Signature...			
	Source Port		Destination Port	
	Protocol	Access Control Policy UUID		
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	AC Pol UUID, cont.			

The File Event Data Block Fields table describes the fields in the file event data block:

File Event Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
File Event Block Type	uint32	Initiates whether file event data block. This value is always 23.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.

File Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — UNKNOWN — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Sourcefire cloud for a disposition. • 5 — NO_CLOUD_RESP — The Sourcefire cloud services did not respond to the request.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.

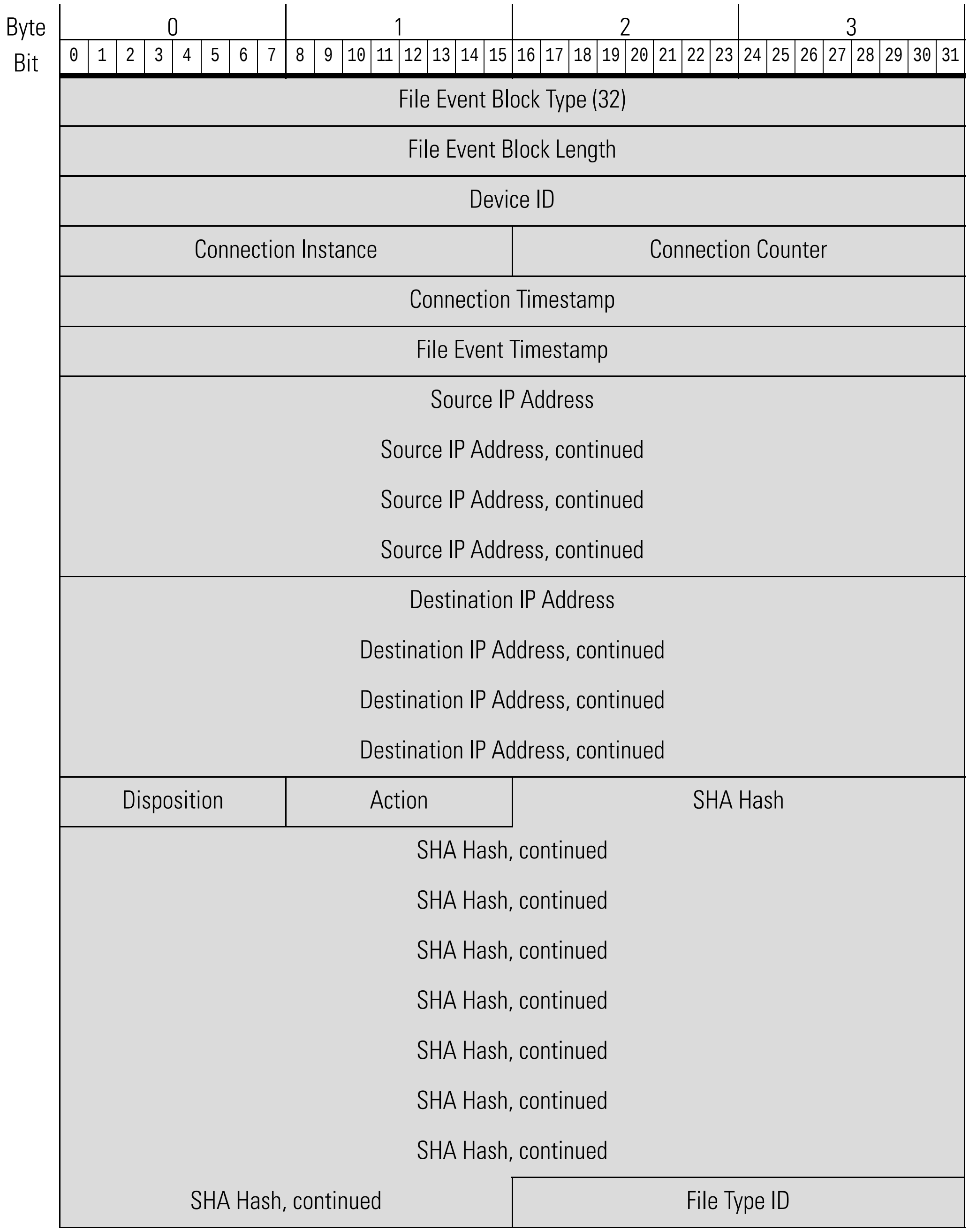
File Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.

File Event for 5.2.x

The file event contains information on files that are sent over the network. This includes the connection information, whether the file is malware, and specific information to identify the file. The file event has a block type of 32 in the series 2 group of blocks. It supersedes block type 23. New fields have been added to track source and destination country, as well as the client and web application instances.

The following graphic shows the structure of the File Event data block:



Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	File Event Block Type (32)			
	File Event Block Length			
	Device ID			
	Connection Instance		Connection Counter	
	Connection Timestamp			
	File Event Timestamp			
	Source IP Address			
	Source IP Address, continued			
	Source IP Address, continued			
	Source IP Address, continued			
	Destination IP Address			
	Destination IP Address, continued			
	Destination IP Address, continued			
	Destination IP Address, continued			
	Disposition	Action	SHA Hash	
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued			
	SHA Hash, continued		File Type ID	

File Name	File Type ID, cont.		String Block Type (0)	
	String Block Type (0), cont.		String Block Length	
	String Block Length, cont.		File Name...	
	File Size			
	File Size, continued			
	Direction	Application ID		
	App ID, cont.	User ID		
URI	User ID, cont.	String Block Type (0)		
	String Block Type (0), cont.	String Block Length		
	String Block Length, cont.	URI...		
Signature	String Block Type (0)			
	String Block Length			
	Signature...			
	Source Port		Destination Port	
	Protocol	Access Control Policy UUID		
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	Access Control Policy UUID, continued			
	AC Pol UUID, cont.	Source Country		Dst. Country
	Dst. Country, cont.	Web Application ID		
	Web App. ID, cont.	Client Application ID		
	Client App. ID, cont.			

The File Event Data Block Fields table describes the fields in the file event data block:

File Event Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
File Event Block Type	uint32	Initiates whether file event data block. This value is always 23.
File Event Block Length	uint32	Total number of bytes in the file event block, including eight bytes for the file event block type and length fields, plus the number of bytes of data that follows.
Device ID	uint32	ID for the device that generated the event.
Connection Instance	uint16	Snort instance on the device that generated the event. Used to link the event with a connection or intrusion event.
Connection Counter	uint16	Value used to distinguish between connection events that happen during the same second.
Connection Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of the associated connection event.
File Event Timestamp	uint32	UNIX timestamp (seconds since 01/01/1970) of when the file type is identified and the file event generated.
Source IP Address	uint8[16]	IPv4 or IPv6 address for the source of the connection.
Destination IP Address	uint8[16]	IPv4 or IPv6 address for the destination of the connection.

File Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Disposition	uint8	The malware status of the file. Possible values include: • 1 — CLEAN — The file is clean and does not contain malware. • 2 — NEUTRAL — It is unknown whether the file contains malware. • 3 — MALWARE — The file contains malware. • 4 — CACHE_MISS — The software was unable to send a request to the Sourcefire cloud for a disposition, or the Sourcefire cloud services did not respond to the request.
Action	uint8	The action taken on the file based on the file type. Can have the following values: • 1 — Detect • 2 — Block • 3 — Malware Cloud Lookup • 4 — Malware Block • 5 — Malware Whitelist
SHA Hash	uint8[32]	SHA-256 hash of the file, in binary format.
File Type ID	uint32	ID number that maps to the file type.
File Name	string	Name of the file.
File Size	uint64	Size of the file in bytes.
Direction	uint8	Value that indicates whether the file was uploaded or downloaded. Can have the following values: • 1 — Download • 2 — Upload Currently the value depends on the protocol (for example, if the connection is HTTP it is a download).
Application ID	uint32	ID number that maps to the application using the file transfer.
User ID	uint32	ID number for the user logged into the destination host, as identified by the system.

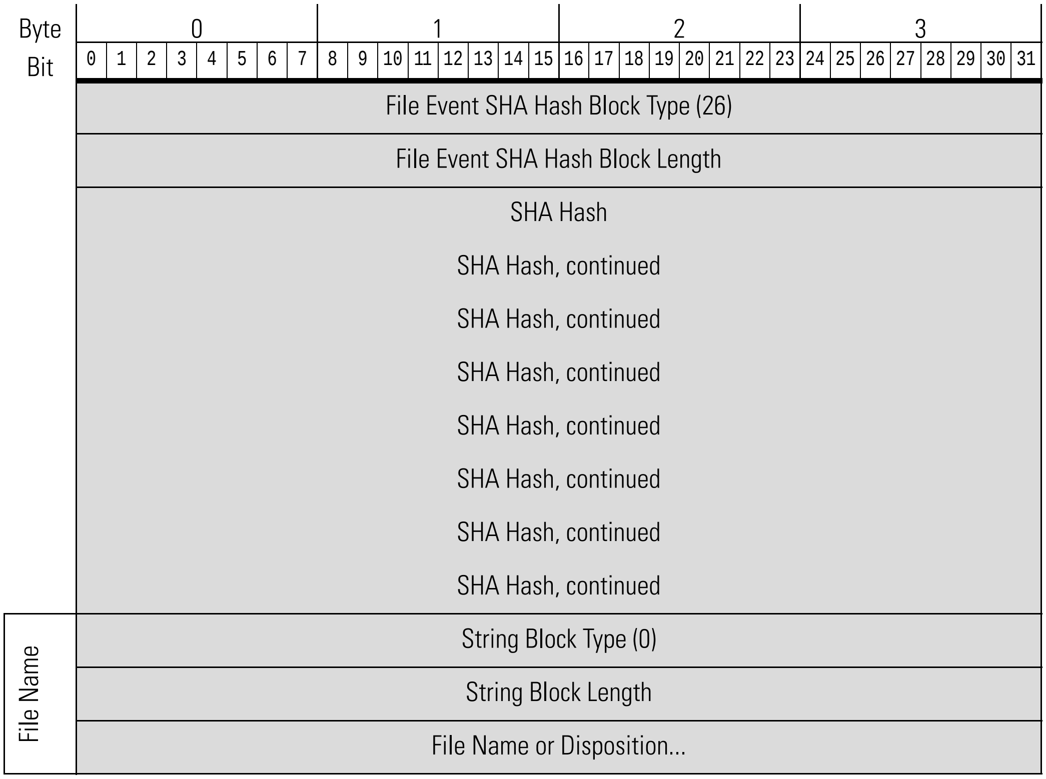
File Event Data Block Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
URI	string	Uniform Resource Identifier (URI) of the connection.
Signature	string	SHA-256 hash of the file, in string format.
Source Port	uint16	Port number for the source of the connection.
Destination Port	uint16	Port number for the destination of the connection.
Protocol	uint8	IANA protocol number specified by the user. For example: • 1 — ICMP • 4 — IP • 6 — TCP • 17 — UDP This is currently only TCP.
Access Control Policy UUID	uint8[16]	Unique identifier for the access control policy that triggered the event.
Source Country	uint16	Code for the country of the source host.
Destination Country	uint16	Code for the country of the destination host.
Web Application ID	uint32	The internal identification number for the web application, if applicable.
Client Application ID	uint32	The internal identification number for the client application, if applicable.

File Event SHA Hash for 5.1.1-5.2.x

The eStreamer service uses the File Event SHA Hash data block to contain metadata of the mapping of the SHA hash of a file to its filename. The block type is 26 in the series 2 list of data blocks. It can be requested if file log events have been requested in the extended requests—event code 111—and either bit 20 is set or metadata is requested with an event version of 4 and an event code of 21.

The following diagram shows the structure of a file event hash data block:



The File Event SHA Hash 5.1.1-5.2.x Data Block Fields table describes the fields in the file event SHA hash data block.

File Event SHA Hash 5.1.1-5.2.x Data Block Fields

FIELD	DATA TYPE	DESCRIPTION
File Event SHA Hash Block Type	uint32	Initiates a File Event SHA Hash block. This value is always 26.
File Event SHA Hash Block Length	uint32	Total number of bytes in the File Event SHA Hash block, including eight bytes for the File Event SHA Hash block type and length fields, plus the number of bytes of data that follows.
SHA Hash	uint8[32]	The SHA-256 hash of the file in binary format.
String Block Type	uint32	Initiates a String data block containing the descriptive name associated with the file. This value is always 0.

File Event SHA Hash 5.1.1-5.2.x Data Block Fields (Continued)

Field	Data Type	Description
String Block Length	uint32	The number of bytes included in the name String data block, including eight bytes for the block type and header fields plus the number of bytes in the Name field.
File Name or Disposition	string	The descriptive name or disposition of the file. If the file is clean, this value is `Clean`. If the file's disposition is unknown, the value is `Neutral`. If the file contains malware, the file name is given.

Legacy Correlation Event Data Structures

The following topics describe other legacy correlation (compliance) data structures:

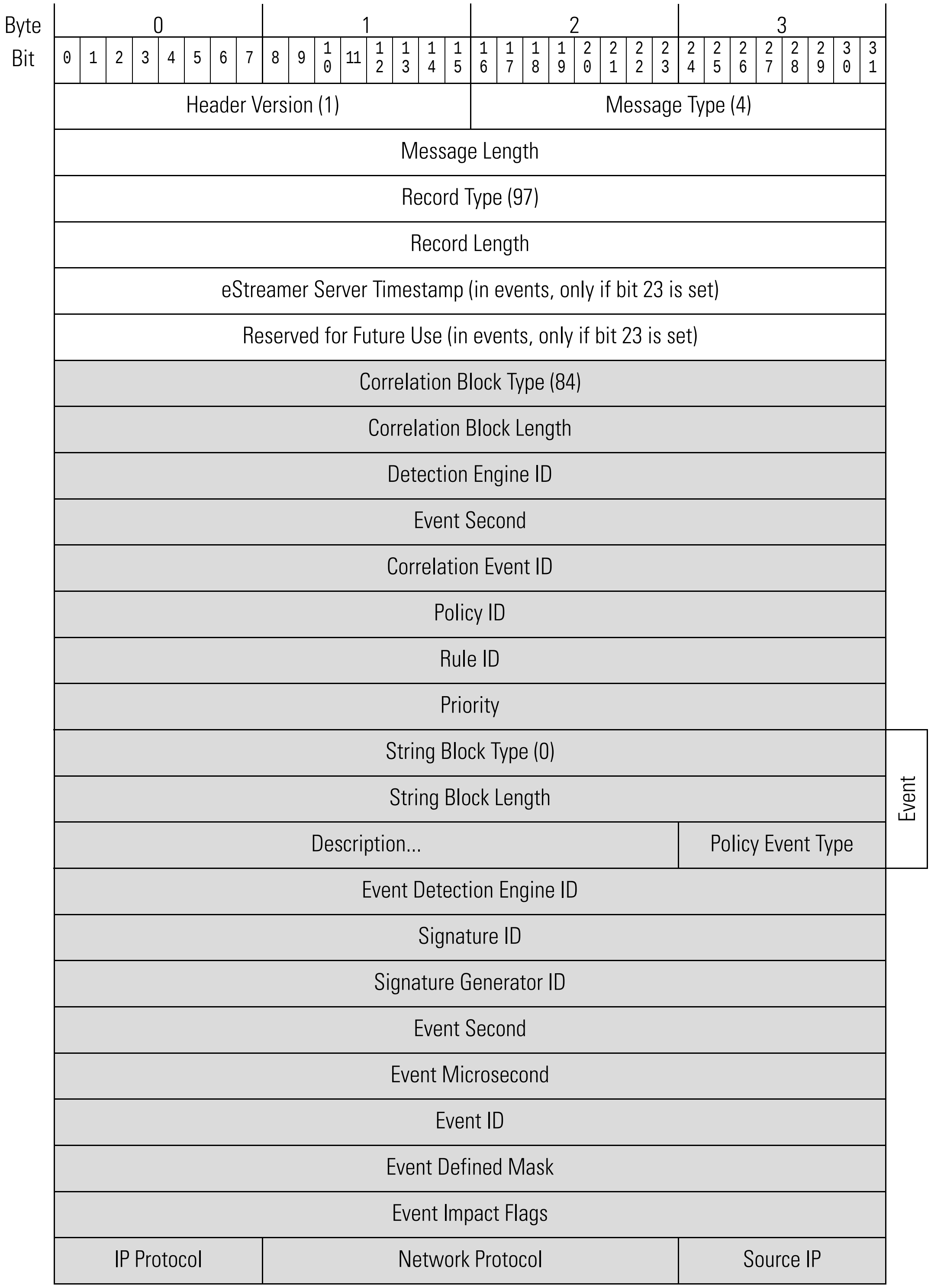
- Correlation Event for 4.8.0.2 - 4.9.1.x on page 630
- Correlation Event for 4.10.x on page 638
- Correlation Event for 5.0 - 5.0.2 on page 646

Correlation Event for 4.8.0.2 - 4.9.1.x

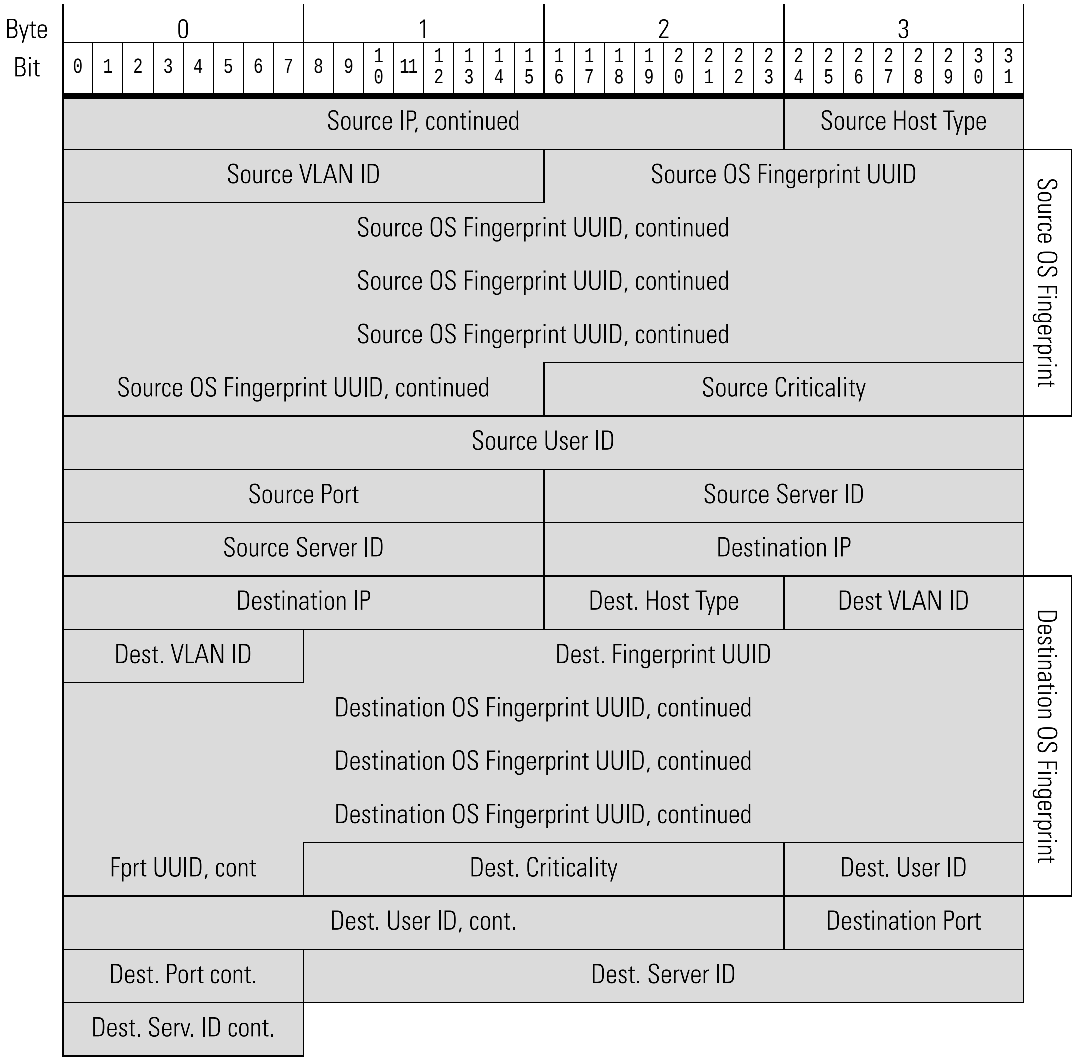
Correlation events contain information about policy violations and are transmitted when correlation policies are violated. The Defense Center uses the standard message header with a record type of 97, followed by a correlation data block with a type of 84. The source and destination user ID fields were added in the 4.7.0.2 - 4.8 version.

You can request that eStreamer transmit 4.8.0.2 - 4.9.1.x correlation events by setting bit 22 in the Flags field of a request message. If you enable bit 23, an extended event header is included in the record.

To request user record metadata along with the policy event data, you must request policy event data using bit 22 and request version 4 metadata (bit 20). For more information, see User Record on page 188.



Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Header Version (1)		Message Type (4)		
	Message Length				
	Record Type (97)				
	Record Length				
	eStreamer Server Timestamp (in events, only if bit 23 is set)				
	Reserved for Future Use (in events, only if bit 23 is set)				
	Correlation Block Type (84)				
	Correlation Block Length				
	Detection Engine ID				
	Event Second				
	Correlation Event ID				
	Policy ID				
	Rule ID				
	Priority				
	String Block Type (0)				Event
	String Block Length				
	Description...			Policy Event Type	
	Event Detection Engine ID				
	Signature ID				
	Signature Generator ID				
	Event Second				
	Event Microsecond				
	Event ID				
	Event Defined Mask				
	Event Impact Flags				
	IP Protocol	Network Protocol		Source IP	



The Correlation Event Data 4.8.0.2 - 4.9.1.x Fields table describes each data field in a correlation event.

Correlation Event Data 4.8.0.2 - 4.9.1.x Fields

FIELD	DATA TYPE	DESCRIPTION
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of 84.
Correlation Block Length	uint32	Length of the correlation data block, that includes 8 bytes for the correlation block type and length plus the correlation data that follows.

Correlation Event Data 4.8.0.2 - 4.9.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Detection Engine ID	uint32	ID of the detection engine or Defense Center that generated the correlation event. A value of zero indicates the Defense Center. You can obtain detection engine names and the detection engine UUIDs that correlate to them by requesting Version 3 metadata. See Detection Engine Record for 4.6.1 - 4.10.x on page 719 for more information.
Event Second	uint32	UNIX timestamp indicating the time that the event was detected (in seconds from 01/01/1970).
Correlation Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Priority	uint32	Priority assigned to the event. This is an integer value from 0 to 5.
String Block Type	uint32	Initiates a string data block that contains the policy violation event description. This value is always set to 0.
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.
Description	string	Description of the correlation event.
Policy Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, discovery, or user event: • 1 — intrusion • 2 — intrusion • 3 — user

Correlation Event Data 4.8.0.2 - 4.9.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Event Detection Engine ID	uint32	Identification number of the detection engine that generated the intrusion or discovery event that triggered the correlation event. You can obtain detection engine IDs and the detection engine UUIDs that correlate to them by requesting Version 3 metadata. See Detection Engine Record for 4.6.1 - 4.10.x on page 719 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is 0.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Sourcefire 3D System preprocessor or rules engine that generated the event.
Event Second	uint32	UNIX timestamp indicating the time that the event was detected (in seconds from 01/01/1970).
Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the device.
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See the Event Defined Values table on page 645 for a list of each bit value.

Correlation Event Data 4.8.0.2 - 4.9.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Event Impact Flags	bits[32]	Impact level of the event. The low-order six bits are used and the impact is determined by how the bits are set. Values are: • 0x00000001 — Source or destination host is in a monitored network monitored (bit 0). • 0x00000002 — Source or destination host exists in the network map (bit 1). • 0x00000004 — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol (bit 2). • 0x00000008 — There is a vulnerability mapped to the operating system of the source or destination host in the event (bit 3). • 0x00000010 — There is a vulnerability mapped to the server detected in the event (bit 4). • 0x00000020 — The event caused the sensor to drop the session (used only when the sensor is running in inline mode) (bit 5). Corresponds to blocked status in Inline Result column in the Sourcefire 3D System web interface. On the Defense Center, the following values map to specific priorities. An `X` indicates that the value can be 0 or 1: • Gray (0, unknown): **`X00000`** • Red (1, vulnerable): **`XX1XXX, X1XXXX`** • Orange (2, potentially vulnerable): **`X00111`** • Yellow (3, currently not vulnerable): **`X00011`** • Blue (4, unknown target): **`X00001`** • Black (dropped packet): **`1XXXXX`**
IP Protocol	uint8	IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP	uint8[4]	IP address of the source host in the event, in IP address octets.
Source Host Type	uint8	Source host's type: • 0 — Host • 1 — Router • 2 — Bridge

Correlation Event Data 4.8.0.2 - 4.9.1.x Fields (Continued)

Field	Data Type	Description
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	IP address of the destination host associated with the policy violation (if applicable). This value will be 0 if there is no destination IP address.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.

Correlation Event Data 4.8.0.2 - 4.9.1.x Fields (Continued)

Field	Data Type	Description
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Server ID	uint32	Identification number for the server running on the source host.

Event Data Mask Field Values

The Event Defined Values table describes each Event Defined Mask value.

Event Defined Values

Description	Mask Value
Event Impact Flags	`0x00000001`
IP Protocol	`0x00000002`
Network Protocol	`0x00000004`
Source IP	`0x00000008`
Source Host Type	`0x00000010`
Source VLAN ID	`0x00000020`
Source Fingerprint ID	`0x00000040`
Source Criticality	`0x00000080`
Source Port	`0x00000100`
Source Server	`0x00000200`

Event Defined Values (Continued)

Description	Mask Value
Destination IP	`0x00000400`
Destination Host Type	`0x00000800`
Destination VLAN ID	`0x00001000`
Destination Fingerprint ID	`0x00002000`
Destination Criticality	`0x00004000`
Destination Port	`0x00008000`
Destination Server	`0x00010000`
Source User	`0x00020000`
Destination User	`0x00040000`

Correlation Event for 4.10.x

Correlation events contain information about policy violations and are transmitted when correlation policies are violated. The Defense Center uses the standard message header with a record type of 112, followed by a correlation data block with a type of 107.

<table>
<tr><td>Byte</td><td>0</td><td>1</td><td>2</td><td>3</td><td></td></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td><td></td></tr>
<tr><td></td><td colspan="2">Header Version (1)</td><td colspan="2">Message Type (4)</td><td></td></tr>
<tr><td></td><td colspan="4">Message Length</td><td></td></tr>
<tr><td></td><td colspan="4">Record Type (112)</td><td></td></tr>
<tr><td></td><td colspan="4">Record Length</td><td></td></tr>
<tr><td></td><td colspan="4">eStreamer Server Timestamp (in events, only if bit 23 is set)</td><td></td></tr>
<tr><td></td><td colspan="4">Reserved for Future Use (in events, only if bit 23 is set)</td><td></td></tr>
<tr><td></td><td colspan="4">Correlation Block Type (107)</td><td></td></tr>
<tr><td></td><td colspan="4">Correlation Block Length</td><td></td></tr>
<tr><td></td><td colspan="4">Detection Engine ID</td><td></td></tr>
<tr><td></td><td colspan="4">Event Second</td><td></td></tr>
<tr><td></td><td colspan="4">Correlation Event ID</td><td></td></tr>
<tr><td></td><td colspan="4">Policy ID</td><td></td></tr>
<tr><td></td><td colspan="4">Rule ID</td><td></td></tr>
<tr><td></td><td colspan="4">Priority</td><td></td></tr>
<tr><td></td><td colspan="4">String Block Type (0)</td><td rowspan="3">Event</td></tr>
<tr><td></td><td colspan="4">String Block Length</td></tr>
<tr><td></td><td colspan="3">Description...</td><td>Event Type</td></tr>
<tr><td></td><td colspan="4">Event Detection Engine ID</td><td></td></tr>
<tr><td></td><td colspan="4">Signature ID</td><td></td></tr>
<tr><td></td><td colspan="4">Signature Generator ID</td><td></td></tr>
<tr><td></td><td colspan="4">Event Second</td><td></td></tr>
<tr><td></td><td colspan="4">Event Microsecond</td><td></td></tr>
<tr><td></td><td colspan="4">Event ID</td><td></td></tr>
<tr><td></td><td colspan="4">Event Defined Mask</td><td></td></tr>
<tr><td></td><td>Event Impact Flags</td><td>IP Protocol</td><td colspan="2">Network Protocol</td><td></td></tr>
<tr><td></td><td colspan="4">Source IP</td><td></td></tr>
</table>

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td><td></td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td></td></tr>
<tr><td></td><td colspan="8">Source Host Type</td><td colspan="16">Source VLAN ID</td><td colspan="8">Source OS Fprt UUID</td><td rowspan="5">Source OS Fprt</td></tr>
<tr><td></td><td colspan="32">Source OS Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="32">Source OS Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="32">Source OS Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="24">Source OS Fingerprint UUID, continued</td><td colspan="8">Source Criticality</td></tr>
<tr><td></td><td colspan="8">Source Criticality, cont</td><td colspan="24">Source User ID</td><td></td></tr>
<tr><td></td><td colspan="8">Source User ID, cont</td><td colspan="16">Source Port</td><td colspan="8">Source Server ID</td><td></td></tr>
<tr><td></td><td colspan="24">Source Server ID, continued</td><td colspan="8">Destination IP</td><td></td></tr>
<tr><td></td><td colspan="24">Destination IP, continued</td><td colspan="8">Dest. Host Type</td><td></td></tr>
<tr><td></td><td colspan="16">Dest. VLAN ID</td><td colspan="16">Destination OS Fingerprint UUID</td><td rowspan="5">Dest OS Fingerprint</td></tr>
<tr><td></td><td colspan="32">Destination OS Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="32">Destination OS Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="32">Destination OS Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="16">Destination OS Fingerprint UUID, continued</td><td colspan="16">Dest. Criticality</td></tr>
<tr><td></td><td colspan="32">Dest. User ID</td><td></td></tr>
<tr><td></td><td colspan="16">Destination Port</td><td colspan="16">Dest. Server ID</td><td></td></tr>
<tr><td></td><td colspan="16">Dest. Server ID, continued</td><td colspan="8">Blocked</td><td colspan="8"></td><td></td></tr>
</table>

The Correlation Event 4.10.x Data Fields table describes each data field in a correlation event.

Correlation Event 4.10.x Data Fields

Field	Data Type	Description
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of 107.
Correlation Block Length	uint32	Length of the correlation data block, which includes 8 bytes for the correlation block type and length plus the correlation data that follows.

Correlation Event 4.10.x Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Detection Engine ID	uint32	ID of the detection engine or Defense Center that generated the correlation event. A value of zero indicates the Defense Center. You can obtain detection engine IDs and the detection engine UUIDs that correlate to them by requesting Version 3 metadata. See Detection Engine Record for 4.6.1 - 4.10.x on page 719 for more information.
Event Second	uint32	UNIX timestamp indicating the time that the event was detected (in seconds from 01/01/1970).
Correlation Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Priority	uint32	Priority assigned to the event. This is an integer value from 0 to 5.
String Block Type	uint32	Initiates a string data block that contains the correlation violation event description. This value is always set to 0.
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.
Description	string	Description of the correlation event.
Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, discovery, or user activity event: • 1 — intrusion • 2 — discovery • 3 — user activity

Correlation Event 4.10.x Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Event Detection Engine ID	uint32	Identification number of the detection engine that generated the intrusion or discovery event that triggered the correlation event. You can obtain detection engine IDs and the detection engine UUIDs that correlate to them by requesting Version 3 metadata. See Detection Engine Record for 4.6.1 - 4.10.x on page 719 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is 0.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Sourcefire 3D System preprocessor or rules engine that generated the event.
Event Second	uint32	UNIX timestamp indicating the time that the event was detected (in seconds from 01/01/1970).
Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the device.
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See the Event Defined Values table on page 645 for a list of each bit value.

Correlation Event 4.10.x Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Event Impact Flags	bits[8]	Impact level of the event. The low-order seven bits are used and the impact is determined by how the bits are set. Values are: • 0x01 — Source or destination host is in a monitored network monitored (bit 0). • 0x02 — Source or destination host exists in the network map (bit 1). • 0x04 — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol (bit 2). • 0x08 — There is a vulnerability mapped to the operating system of the source or destination host in the event (bit 3). • 0x10 — There is a vulnerability mapped to the server detected in the event (bit 4). • 0x20 — The event caused the sensor to drop the session (used only when the sensor is running in inline mode) (bit 5). Corresponds to blocked status in Inline Result column in the Sourcefire 3D System web interface. • 0x40 — The rule that generated this event contains rule metadata setting the impact flag to red (bit 6). If the rule is provided by the Sourcefire Vulnerability Research Team (VRT), the source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. On the Defense Center, the following values map to specific priorities. An `X` indicates that the value can be 0 or 1: • Gray (0, unknown): **`0X00000`** • Red (1, vulnerable): **`XXX1XXX, XX1XXXX, 1XXXXXX`** • Orange (2, potentially vulnerable): **`0X00111`** • Yellow (3, currently not vulnerable): **`0X00011`** • Blue (4, unknown target): **`0X00001`** • Black (dropped packet): **`X1XXXXX`**
IP Protocol	uint8	IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP	uint8[4]	IP address of the source host in the event, in IP address octets.

Correlation Event 4.10.x Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Source Host Type	uint8	Source host's type: • 0 — Host • 1 — Router • 2 — Bridge
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	IP address of the destination host associated with the policy violation (if applicable). This value will be 0 if there is no destination IP address.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.

Correlation Event 4.10.x Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Server ID	uint32	Identification number for the server running on the source host.
Blocked	uint8	Value indicating what happened to the packet that triggered the intrusion event. • 0 — Intrusion event not dropped • 1 — Intrusion event was dropped (inline mode, drop when inline is set) • 2 — The packet that triggered the event would have been dropped, if the intrusion policy had been applied to a detection engine using an inline interface set.

Event Data Mask Field Values

The Event Defined Values table describes each value in the Event Defined Mask.

Event Defined Values

DESCRIPTION	MASK VALUE
Event Impact Flags	0x00000001
IP Protocol	0x00000002

Event Defined Values (Continued)

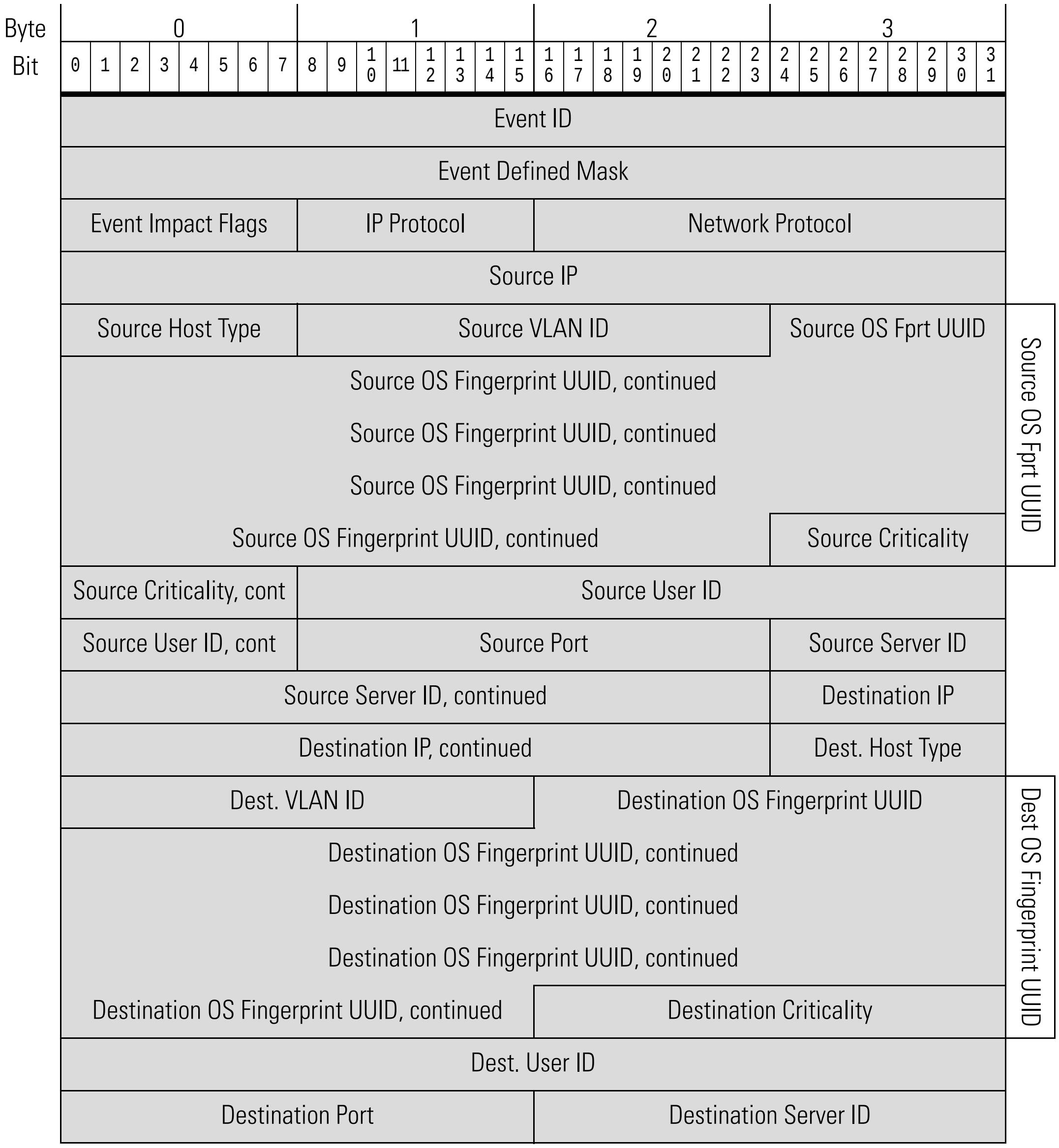
DESCRIPTION	MASK VALUE
Network Protocol	0x00000004
Source IP	0x00000008
Source Host Type	0x00000010
Source VLAN ID	0x00000020
Source Fingerprint ID	0x00000040
Source Criticality	0x00000080
Source Port	0x00000100
Source Server	0x00000200
Destination IP	0x00000400
Destination Host Type	0x00000800
Destination VLAN ID	0x00001000
Destination Fingerprint ID	0x00002000
Destination Criticality	0x00004000
Destination Port	0x00008000
Destination Server	0x00010000
Source User	0x00020000
Destination User	0x00040000

Correlation Event for 5.0 - 5.0.2

Correlation events (called compliance events in pre-5.0 versions) contain information about correlation policy violations. This message uses the standard eStreamer message header and specifies a record type of 112, followed by a correlation data block of type 116. Data block type 116 differs from its predecessor (block type 107) in including additional information about the associated security zone and interface.

You can request 5.0+ correlation events from eStreamer only by extended request, for which you request event type code 31 and version code 7 in the Stream Request message (see Submitting Extended Requests on page 20 for information about submitting extended requests). You can optionally enable bit 23 in the flags field of the initial event stream request message, to include the extended event header. You can also enable bit 20 in the flags field to include user metadata.

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Header Version (1)		Message Type (4)		
	Message Length				
	Record Type (112)				
	Record Length				
	eStreamer Server Timestamp (in events, only if bit 23 is set)				
	Reserved for Future Use (in events, only if bit 23 is set)				
	Correlation Block Type (116)				
	Correlation Block Length				
	Device ID				
	(Correlation) Event Second				
	Event ID				
	Policy ID				
	Rule ID				
	Priority				
	String Block Type (0)				Event
	String Block Length				
	Description...			Event Type	
	Event Device ID				
	Signature ID				
	Signature Generator ID				
	(Trigger) Event Second				
	(Trigger) Event Microsecond				



Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Event ID				
	Event Defined Mask				
	Event Impact Flags	IP Protocol	Network Protocol		
	Source IP				
	Source Host Type	Source VLAN ID		Source OS Fprt UUID	Source OS Fprt UUID
	Source OS Fingerprint UUID, continued				
	Source OS Fingerprint UUID, continued				
	Source OS Fingerprint UUID, continued				
	Source OS Fingerprint UUID, continued			Source Criticality	
	Source Criticality, cont	Source User ID			
	Source User ID, cont	Source Port		Source Server ID	
	Source Server ID, continued			Destination IP	
	Destination IP, continued			Dest. Host Type	
	Dest. VLAN ID		Destination OS Fingerprint UUID		Dest OS Fingerprint UUID
	Destination OS Fingerprint UUID, continued				
	Destination OS Fingerprint UUID, continued				
	Destination OS Fingerprint UUID, continued				
	Destination OS Fingerprint UUID, continued		Destination Criticality		
	Dest. User ID				
	Destination Port		Destination Server ID		

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Destination Server ID, cont.		Blocked	Ingress Interface UUID
	Ingress Interface UUID, continued			
	Ingress Interface UUID, continued			
	Ingress Interface UUID, continued			
	Ingress Interface UUID, continued			Egress Interface UUID
	Egress Interface UUID, continued			
	Egress Interface UUID, continued			
	Egress Interface UUID, continued			
	Egress Interface UUID, continued			Ingress Zone UUID
	Ingress Zone UUID			
	Ingress Zone UUID, continued			
	Ingress Zone UUID, continued			
	Ingress Zone UUID, continued			Egress Zone UUID
	Egress Zone UUID			
	Egress Zone UUID, continued			
	Egress Zone UUID, continued			
	Egress Zone UUID, continued			

Note that the record structure includes a String block type, which is a block in series 1. For information about series 1 blocks, see Understanding Discovery (Series 1) Blocks on page 224.

Correlation Event 5.0 - 5.0.2 Data Fields

FIELD	DATA TYPE	DESCRIPTION
Correlation Block Type	uint32	Indicates a correlation event data block follows. This field always has a value of 107. See Understanding Discovery (Series 1) Blocks on page 224.
Correlation Block Length	uint32	Length of the correlation data block, which includes 8 bytes for the correlation block type and length plus the correlation data that follows.
Device ID	uint32	Internal identification number of the managed device or Defense Center that generated the correlation event. A value of zero indicates the Defense Center. You can obtain managed device names by requesting Version 3 metadata. See Managed Device Record Metadata on page 99 for more information.
(Correlation) Event Second	uint32	UNIX timestamp indicating the time that the correlation event was generated (in seconds from 01/01/1970).
Event ID	uint32	Correlation event identification number.
Policy ID	uint32	Identification number of the correlation policy that was violated. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Rule ID	uint32	Identification number of the correlation rule that triggered to violate the policy. See Server Record on page 182 for information about how to obtain policy identification numbers from the database.
Priority	uint32	Priority assigned to the event. This is an integer value from 0 to 5.
String Block Type	uint32	Initiates a string data block that contains the correlation violation event description. This value is always set to 0. For more information about string blocks, see String Data Block on page 237.
String Block Length	uint32	Number of bytes in the event description string block, which includes four bytes for the string block type and four bytes for the string block length, plus the number of bytes in the description.

Correlation Event 5.0 - 5.0.2 Data Fields (Continued)

Field	Data Type	Description
Description	string	Description of the correlation event.
Event Type	uint8	Indicates whether the correlation event was triggered by an intrusion, host discovery, or user event: • 1 — intrusion • 2 — host discovery • 3 — user
Event Device ID	uint32	Identification number of the device that generated the event that triggered the correlation event. You can obtain device name by requesting Version 3 metadata. See Managed Device Record Metadata on page 99 for more information.
Signature ID	uint32	If the event was an intrusion event, indicates the rule identification number that corresponds with the event. Otherwise, the value is 0.
Signature Generator ID	uint32	If the event was an intrusion event, indicates the ID number of the Sourcefire 3D System preprocessor or rules engine that generated the event.
(Trigger) Event Second	uint32	UNIX timestamp indicating the time of the event that triggered the correlation policy rule (in seconds from 01/01/1970).
(Trigger) Event Microsecond	uint32	Microsecond (one millionth of a second) increment that the event was detected.
Event ID	uint32	Identification number of the event generated by the device.
Event Defined Mask	bits[32]	Set bits in this field indicate which of the fields that follow in the message are valid. See the Event Defined Values table on page 655 for a list of each bit value.

Correlation Event 5.0 - 5.0.2 Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Event Impact Flags	bits[8]	Impact flag value of the event. The low-order eight bits indicate the impact level. Values are: • 0x01 (bit 0) — Source or destination host is in a network monitored by the system. • 0x02 (bit 1) — Source or destination host exists in the network map. • 0x04 (bit 2) — Source or destination host is running a server on the port in the event (if TCP or UDP) or uses the IP protocol. • 0x08 (bit 3) — There is a vulnerability mapped to the operating system of the source or destination host in the event. • 0x10 (bit 4) — There is a vulnerability mapped to the server detected in the event. • 0x20 (bit 5) — The event caused the managed device to drop the session (used only when the device is running in inline, switched, or routed deployment). Corresponds to blocked status in the Sourcefire 3D System web interface. • 0x40 (bit 6) — The rule that generated this event contains rule metadata setting the impact flag to red (bit 6). The source or destination host is potentially compromised by a virus, trojan, or other piece of malicious software. • 0x80 (bit 7) — There is a vulnerability mapped to the client detected in the event. The following impact level values map to specific priorities on the Defense Center. An `X` indicates the value can be 0 or 1: • gray (0, unknown): `00X00000` • red (1, vulnerable): `XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX` • orange (2, potentially vulnerable): `00X00111` • yellow (3, currently not vulnerable): `00X00011` • blue (4, unknown target): `00X00001`
IP Protocol	uint8	Identifier of the IP protocol associated with the event, if applicable.
Network Protocol	uint16	Network protocol associated with the event, if applicable.
Source IP	uint8[4]	IP address of the source host in the event, in IP address octets.

Correlation Event 5.0 - 5.0.2 Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Source Host Type	uint8	Source host's type: • 0 — Host • 1 — Router • 2 — Bridge
Source VLAN ID	uint16	Source host's VLAN identification number, if applicable.
Source OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts a unique identifier for the source host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Source Criticality	uint16	User-defined criticality value for the source host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Source User ID	uint32	Identification number for the user logged into the source host, as identified by the system.
Source Port	uint16	Source port in the event.
Source Server ID	uint32	Identification number for the server running on the source host.
Destination IP Address	uint8[4]	IP address of the destination host associated with the policy violation (if applicable). This value will be 0 if there is no destination IP address.
Destination Host Type	uint8	Destination host's type: • 0 — Host • 1 — Router • 2 — Bridge
Destination VLAN ID	uint16	Destination host's VLAN identification number, if applicable.

Correlation Event 5.0 - 5.0.2 Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Destination OS Fingerprint UUID	uint8[16]	A fingerprint ID number that acts as a unique identifier for the destination host's operating system. See Server Record on page 182 for information about obtaining the values that map to the fingerprint IDs.
Destination Criticality	uint16	User-defined criticality value for the destination host: • 0 — None • 1 — Low • 2 — Medium • 3 — High
Destination User ID	uint32	Identification number for the user logged into the destination host, as identified by the system.
Destination Port	uint16	Destination port in the event.
Destination Service ID	uint32	Identification number for the server running on the source host.
Blocked	uint8	Value indicating what happened to the packet that triggered the intrusion event. • 0 — Intrusion event not dropped • 1 — Intrusion event was dropped (drop when deployment is inline, switched, or routed) • 2 — The packet that triggered the event would have been dropped, if the intrusion policy had been applied to a device in inline, switched, or routed deployment.
Ingress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the ingress interface associated with correlation event.
Egress Interface UUID	uint8[16]	An interface ID that acts as the unique identifier for the egress interface associated with correlation event.

Correlation Event 5.0 - 5.0.2 Data Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Ingress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the ingress security zone associated with correlation event.
Egress Zone UUID	uint8[16]	A zone ID that acts as the unique identifier for the egress security zone associated with correlation event.

The Event Defined Values table describes each Event Defined Mask value.

Event Defined Values

DESCRIPTION	MASK VALUE
Event Impact Flags	`0x00000001`
IP Protocol	`0x00000002`
Network Protocol	`0x00000004`
Source IP	`0x00000008`
Source Host Type	`0x00000010`
Source VLAN ID	`0x00000020`
Source Fingerprint ID	`0x00000040`
Source Criticality	`0x00000080`
Source Port	`0x00000100`
Source Server	`0x00000200`
Destination IP	`0x00000400`
Destination Host Type	`0x00000800`
Destination VLAN ID	`0x00001000`
Destination Fingerprint ID	`0x00002000`
Destination Criticality	`0x00004000`

Event Defined Values (Continued)

DESCRIPTION	MASK VALUE
Destination Port	`0x00008000`
Destination Server	`0x00010000`
Source User	`0x00020000`
Destination User	`0x00040000`

Legacy Host Data Structures

To request these structures, you must use a Host Request Message. To request a legacy structure, the Host Request Message must use an older format. See Host Request Message Format on page 47 for more information.

The following topics describe legacy host data structures, including both host profile and full host profile structures:

- Full Host Profile Data Block 4.8 on page 656
- Full Host Profile Data Block 4.9 - 4.10.x on page 662
- Full Host Profile Data Block 5.0 - 5.0.2 on page 673
- Full Host Profile Data Block 5.1.1 on page 685
- Full Host Profile Data Block 5.2.x on page 696
- Host Profile Data Block for 5.1.x on page 711
- IP Range Specification Data Block for 4.7.x - 5.1.1.x on page 718

Full Host Profile Data Block 4.8

The Full Host Profile data block contains a full set of data describing one host. The eStreamer server generates and transmits Full Host Profile data blocks in host request data messages, which it sends in response to host request messages submitted by the client. The full host profile data block for 4.8 has the format shown in the following graphic. Note that the graphic shows all fields in the record, but the content details of nested data blocks are omitted. For information about the fields in the encapsulated blocks, see the subsections of this guide that described the data block in question. The Full Host Profile Data Block for version 4.8 has a data block type value of 47.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

<table>
<tr><th>Byte</th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td></tr>
<tr><td></td><td colspan="4">Full Host Profile Data Block (47)</td></tr>
<tr><td></td><td colspan="4">Data Block Length</td></tr>
<tr><td></td><td colspan="4">IP Address</td></tr>
<tr><td></td><td>Hops</td><td colspan="3">Confidence</td></tr>
<tr><td></td><td>Confidence</td><td colspan="3">Fingerprint UUID</td></tr>
<tr><td></td><td colspan="4">Fingerprint UUID, continued</td></tr>
<tr><td></td><td colspan="4">Fingerprint UUID, continued</td></tr>
<tr><td></td><td>Fingerprint UUID, cont.</td><td colspan="3">List Block Type (11)...</td></tr>
<tr><td></td><td>List Block Type (11)...</td><td colspan="3">List Block Length...</td></tr>
<tr><td></td><td>List Block Length...</td><td colspan="3">(TCP) Full Server Data Blocks *</td></tr>
<tr><td></td><td colspan="4">List Block Type (11)</td></tr>
<tr><td></td><td colspan="4">List Block Length</td></tr>
<tr><td></td><td colspan="4">(UDP) Full Server Data Blocks *</td></tr>
<tr><td></td><td colspan="4">List Block Type (11)</td></tr>
<tr><td></td><td colspan="4">List Block Length</td></tr>
<tr><td></td><td colspan="4">(Network) Protocol Data Blocks *</td></tr>
<tr><td></td><td colspan="4">List Block Type (11)</td></tr>
<tr><td></td><td colspan="4">List Block Length</td></tr>
<tr><td></td><td colspan="4">(Transport) Protocol Data Blocks *</td></tr>
<tr><td></td><td colspan="4">List Block Type (11)</td></tr>
<tr><td></td><td colspan="4">List Block Length</td></tr>
<tr><td></td><td colspan="4">Host MAC Address Data Blocks *</td></tr>
<tr><td></td><td colspan="4">Last Seen</td></tr>
<tr><td></td><td colspan="4">Host Type</td></tr>
<tr><td></td><td colspan="2">Business Criticality</td><td colspan="2">VLAN ID</td></tr>
<tr><td></td><td>VLAN Type</td><td>VLAN Priority</td><td colspan="2">Generic List Block Type (31)</td></tr>
</table>

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Generic List Block Type, continued		Generic List Block Length	
	Generic List Block Length, continued		Client Application Data Blocks *	
NetBIOS Name	String Block Type (0)			
	String Block Length			
	NetBIOS Name String...			
Notes Data	String Block Type (0)			
	String Block Length			
	Notes String....			
	Generic List Block Type (31)			
	Generic List Block Length			
	(VDB) Host Vulnerability Data Blocks *			
	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party Scan) Host Vulnerability Data Blocks *			
	List Block Type (11)			
	List Block Length			
	Attribute Value Data Blocks *			

The Full Host Profile Data Block 4.8 table describes the components of the Full Host Profile record.

Full Host Profile Data Block 4.8

FIELD	DATA TYPE	DESCRIPTION
IP Address	uint8[4]	IP address of the host, in IP address octets.
Hops	uint8	Number of network hops from the host to the detection device.
Confidence	uint32	Percentage of confidence of Sourcefire in correct identification of the host data.

Full Host Profile Data Block 4.8 (Continued)

Field	Data Type	Description
Fingerprint UUID	uint8[16]	UUID of the OS Fingerprint
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of MAC Address data blocks. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.

Full Host Profile Data Block 4.8 (Continued)

Field	Data Type	Description
Host Type	uint32	Indicates host type. Values include: • 0 — host • 1 — router • 2 — bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Client Application Data Blocks *	variable	List of Client Application data blocks. See Host Client Application Data Block for 4.9.1 - 4.10.x on page 539 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.

Full Host Profile Data Block 4.8 (Continued)

FIELD	DATA TYPE	DESCRIPTION
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Sourcefire vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified through a third party scanner. Note that the host vulnerability IDs for these data blocks are third party scanner IDs, not Sourcefire vulnerability IDs. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.

Full Host Profile Data Block 4.8 (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block on page 253 for a description of this data block.

Full Host Profile Data Block 4.9 - 4.10.x

The Full Host Profile data block contains a full set of data describing one host. The eStreamer server generates and transmits Full Host Profile data blocks in host request data messages, which it sends in response to host request messages submitted by the client. The full host profile data block for 4.9 - 4.10.x has the format shown in the following graphic. Note that the graphic shows all fields in the record, but the content details of nested data blocks are omitted. For information about the fields in the encapsulated blocks, see the subsections of this guide that described the data block in question. The Full Host Profile Data Block for version 4.9 to 4.10.x has a data block type value of 92.

IMPORTANT! An asterisk(*) next to a data block name in the following diagram indicates that multiple instances of the data block may occur.

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Host Profile Data Block (92)			
	Data Block Length			
	IP Address			
	Hops	Generic List Block Type (31)		
	Generic List Block Type, continued	Generic List Block Length		

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Derived Fingerprints	Generic List Block Length, continued	Operating System Fingerprint Block Type (87)*		
	OS Fingerprint Block Type (87)*, con't	Operating System Fingerprint Block Length		
	OS Fingerprint Block Length, con't	Operating System Derived Fingerprint Data...		
	Generic List Block Type (31)			
	Generic List Block Length			
Server Fingerprints	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System Server Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Client Fingerprints	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System Client Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System SMB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System DHCP Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
User Fingerprints	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System User Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Scan Fingerprints	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System Scan Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Application Fingerprints	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System Application Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Conflict Fingerprints	Operating System Fingerprint Block Type (87)*			
	Operating System Fingerprint Block Length			
	Operating System Conflict Fingerprint Data...			
	List Block Type (11)...			
	List Block Length...			
	(TCP) Full Server Data Blocks *			
	List Block Type (11)			
	List Block Length			
	(UDP) Full Server Data Blocks *			
	List Block Type (11)			
	List Block Length			

<table>
<tr><td></td><td>Byte</td><td>0</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td></td><td>Bit</td><td>0 1 2 3 4 5 6 7</td><td>8 9 10 11 12 13 14 15</td><td>16 17 18 19 20 21 22 23</td><td>24 25 26 27 28 29 30 31</td></tr>
<tr><td></td><td></td><td colspan="4">(Network) Protocol Data Blocks *</td></tr>
<tr><td></td><td></td><td colspan="4">List Block Type (11)</td></tr>
<tr><td></td><td></td><td colspan="4">List Block Length</td></tr>
<tr><td></td><td></td><td colspan="4">(Transport) Protocol Data Blocks *</td></tr>
<tr><td></td><td></td><td colspan="4">List Block Type (11)</td></tr>
<tr><td></td><td></td><td colspan="4">List Block Length</td></tr>
<tr><td></td><td></td><td colspan="4">Host MAC Address Data Blocks *</td></tr>
<tr><td></td><td></td><td colspan="4">Last Seen</td></tr>
<tr><td></td><td></td><td colspan="4">Host Type</td></tr>
<tr><td></td><td></td><td colspan="2">Business Criticality</td><td colspan="2">VLAN ID</td></tr>
<tr><td></td><td></td><td>VLAN Type</td><td>VLAN Priority</td><td colspan="2">Generic List Block Type (31)</td></tr>
<tr><td></td><td></td><td colspan="2">Generic List Block Type, continued</td><td colspan="2">Generic List Block Length</td></tr>
<tr><td></td><td></td><td colspan="2">Generic List Block Length, continued</td><td colspan="2">Client Application Data Blocks *</td></tr>
<tr><td rowspan="3">NetBIOS Name</td><td></td><td colspan="4">String Block Type (0)</td></tr>
<tr><td></td><td colspan="4">String Block Length</td></tr>
<tr><td></td><td colspan="4">NetBIOS Name String...</td></tr>
<tr><td rowspan="3">Notes Data</td><td></td><td colspan="4">String Block Type (0)</td></tr>
<tr><td></td><td colspan="4">String Block Length</td></tr>
<tr><td></td><td colspan="4">Notes String....</td></tr>
<tr><td></td><td></td><td colspan="4">Generic List Block Type (31)</td></tr>
<tr><td></td><td></td><td colspan="4">Generic List Block Length</td></tr>
<tr><td></td><td></td><td colspan="4">(VDB) Host Vulnerability Data Blocks *</td></tr>
<tr><td></td><td></td><td colspan="4">Generic List Block Type (31)</td></tr>
<tr><td></td><td></td><td colspan="4">Generic List Block Length</td></tr>
<tr><td></td><td></td><td colspan="4">(Third Party/VDB) Host Vulnerability Data Blocks *</td></tr>
<tr><td></td><td></td><td colspan="4">Generic List Block Type (31)</td></tr>
</table>

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Generic List Block Length			
	(Third Party Scan) Host Vulnerability Data Blocks *			
	List Block Type (11)			
	List Block Length			
	Attribute Value Data Blocks *			

The Full Host Profile Data Block 4.9 - 4.10.x table describes the components of the Full Host Profile record.

Full Host Profile Data Block 4.9 - 4.10.x

FIELD	DATA TYPE	DESCRIPTION
IP Address	uint8[4]	IP address of the host, in IP address octets.
Hops	uint8	Number of network hops from the host to the detection device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Derived Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block for 4.9.x - 5.0.2 on page 575 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP services on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block on page 243 for a description of this data block.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

Field	Data Type	Description
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of MAC Address data blocks. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — host • 1 — router • 2 — bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Client Application Data Blocks *	variable	List of Client Application data blocks. See Host Client Application Data Block for 4.9.1 - 4.10.x on page 539 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

Field	Data Type	Description
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Sourcefire vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.

Full Host Profile Data Block 4.9 - 4.10.x (Continued)

FIELD	DATA TYPE	DESCRIPTION
(Third Party Scan/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified through a third party scanner. Note that the host vulnerability IDs for these data blocks are third party scanner IDs, not Sourcefire vulnerability IDs. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block on page 253 for a description of this data block.

Full Host Profile Data Block 5.0 - 5.0.2

The Full Host Profile data block for version 5.0 - 5.0.2 contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data

Structures on page 164. The Full Host Profile data block a block type value of 111.

IMPORTANT! An asterisk(*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		Full Host Profile Data Block (111)			
		Data Block Length			
		IP Address			
		Hops	Generic List Block Type (31)		
		Generic List Block Type, continued	Generic List Block Length		
OS Derived Fingerprints		Generic List Block Length, continued	Operating System Fingerprint Block Type (130)*		
		OS Fingerprint Block Type (130)*, con't	Operating System Fingerprint Block Length		
		OS Fingerprint Block Length, con't	Operating System Derived Fingerprint Data...		
		Generic List Block Type (31)			
		Generic List Block Length			
Server Fingerprints		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System Server Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			
Client Fingerprints		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System Client Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System VDB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System VDB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
User Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System User Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Scan Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Scan Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Application Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Application Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Conflict Fingerprint Data...			
(TCP) Full Server Data	List Block Type (11)...			
	List Block Length...			
	(TCP) Full Server Data Blocks (104)*			
(UDP) Full Server Data	List Block Type (11)			
	List Block Length			
	(UDP) Full Server Data Blocks (104)*			
Network Protocol Data	List Block Type (11)			
	List Block Length			
	(Network) Protocol Data Blocks (4)*			
Transport Protocol Data	List Block Type (11)			
	List Block Length			
	(Transport) Protocol Data Blocks (4)*			
MAC Address Data	List Block Type (11)			
	List Block Length			
	Host MAC Address Data Blocks (95)*			
	Last Seen			
	Host Type			
	Business Criticality		VLAN ID	
	VLAN Type	VLAN Priority	Generic List Block Type (31)	
Host Client Data	Generic List Block Type, continued		Generic List Block Length	
	Generic List Block Length, continued		Full Host Client Application Data Blocks (112)*	

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
NetBIOS Name	String Block Type (0)			
	String Block Length			
	NetBIOS Name String...			
Notes Data	String Block Type (0)			
	String Block Length			
	Notes String....			
(VDB) Host Vulns	Generic List Block Type (31)			
	Generic List Block Length			
	(VDB) Host Vulnerability Data Blocks (85)*			
3rd Pty/VDB) Host Vulns	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party/VDB) Host Vulnerability Data Blocks (85)*			
3rd Pty Scan Host Vulns	Generic List Block Type (31)			
	Generic List Block Length			
	(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*			
Attribute Value Data	List Block Type (11)			
	List Block Length			
	Attribute Value Data Blocks *			

The Full Host Profile Record 5.0 - 5.0.2 Fields table describes the components of the Full Host Profile for 5.0 - 5.0.2record.

Full Host Profile Record 5.0 - 5.0.2 Fields

Field	Data Type	Description
IP Address	uint8[4]	IP address of the host, in IP address octets.
Hops	uint8	Number of network hops from the host to the device.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

Field	Data Type	Description
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — host • 1 — router • 2 — bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+ on page 331 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.

Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

Field	Data Type	Description
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/ VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Sourcefire-detected IDs. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.

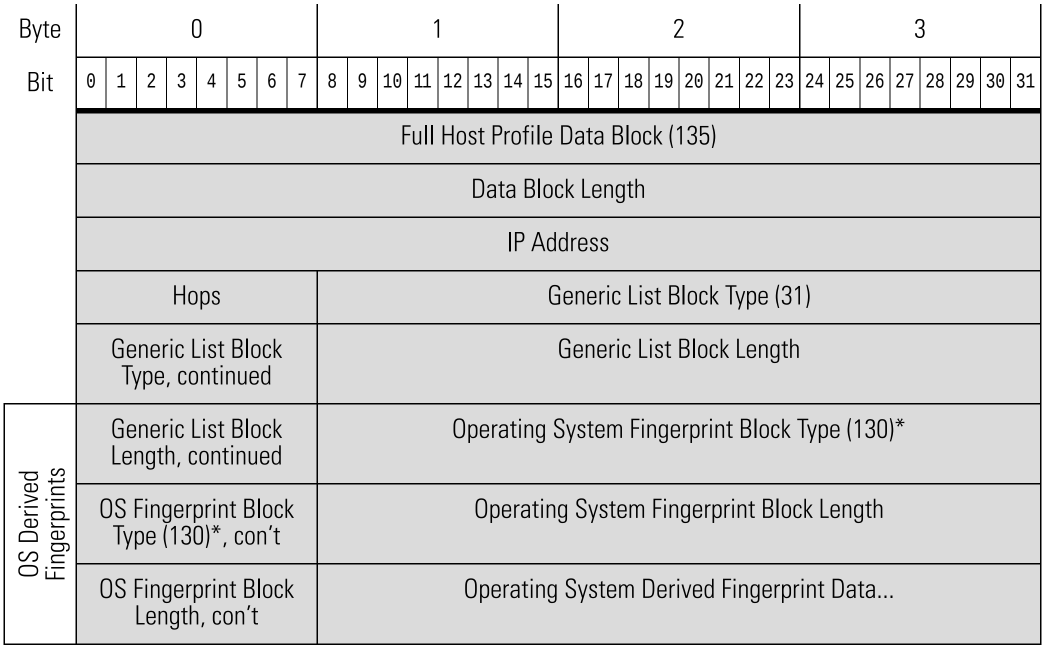
Full Host Profile Record 5.0 - 5.0.2 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block on page 253 for a description of the data blocks in this list.

Full Host Profile Data Block 5.1.1

The Full Host Profile data block for version 5.1.1 contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures on page 164. The Full Host Profile data block a block type value of 135 It deprecates data block 111.

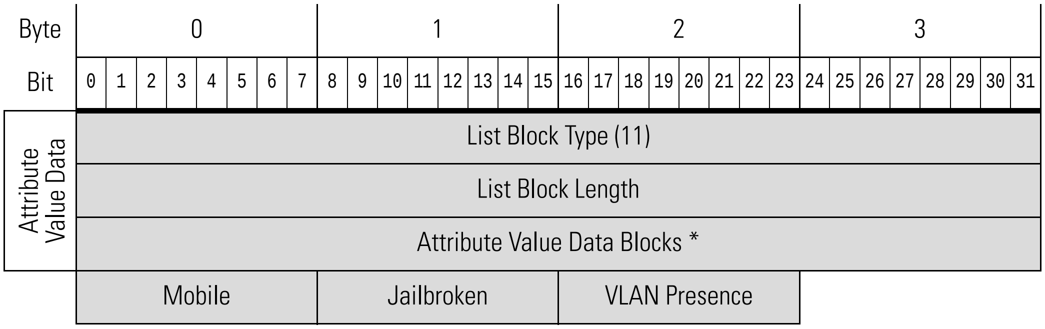
IMPORTANT! An asterisk(*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Full Host Profile Data Block (135)
Data Block Length
IP Address
Hops
Generic List Block Type (31)
Generic List Block Type, continued
Generic List Block Length
OS Derived Fingerprints
Generic List Block Length, continued
Operating System Fingerprint Block Type (130)*
OS Fingerprint Block Type (130)*, con't
Operating System Fingerprint Block Length
OS Fingerprint Block Length, con't
Operating System Derived Fingerprint Data...

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Generic List Block Type (31)			
	Generic List Block Length			
Server Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Server Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Client Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Client Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 1	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System VDB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
VDB Native Fingerprints 2	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System VDB Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
User Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System User Fingerprint Data...			
	Generic List Block Type (31)			

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Generic List Block Length			
Scan Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Scan Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Application Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Application Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
Conflict Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Conflict Fingerprint Data...			
(TCP) Full Server Data	List Block Type (11)...			
	List Block Length...			
	(TCP) Full Server Data Blocks (104)*			
(UDP) Full Server Data	List Block Type (11)			
	List Block Length			
	(UDP) Full Server Data Blocks (104)*			
Network Protocol Data	List Block Type (11)			
	List Block Length			
	(Network) Protocol Data Blocks (4)*			
Transport Protocol Data	List Block Type (11)			
	List Block Length			
	(Transport) Protocol Data Blocks (4)*			

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
MAC Address Data		List Block Type (11)			
		List Block Length			
		Host MAC Address Data Blocks (95)*			
		Last Seen			
		Host Type			
		Business Criticality		VLAN ID	
		VLAN Type	VLAN Priority	Generic List Block Type (31)	
Host Client Data		Generic List Block Type, continued		Generic List Block Length	
		Generic List Block Length, continued		Full Host Client Application Data Blocks (112)*	
NetBIOS Name		String Block Type (0)			
		String Block Length			
		NetBIOS Name String...			
Notes Data		String Block Type (0)			
		String Block Length			
		Notes String....			
(VDB) Host Vulns		Generic List Block Type (31)			
		Generic List Block Length			
		(VDB) Host Vulnerability Data Blocks (85)*			
3rd Pty/VDB) Host Vulns		Generic List Block Type (31)			
		Generic List Block Length			
		(Third Party/VDB) Host Vulnerability Data Blocks (85)*			
3rd Pty Scan Host Vulns		Generic List Block Type (31)			
		Generic List Block Length			
		(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*			



The Full Host Profile Record 5.1.1 Fields table describes the components of the Full Host Profile for 5.1.1 record.

Full Host Profile Record 5.1.1 Fields

FIELD	DATA TYPE	DESCRIPTION
IP Address	uint8[4]	IP address of the host, in IP address octets.
Hops	uint8	Number of network hops from the host to the device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Full Host Profile Record 5.1.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.

Full Host Profile Record 5.1.1 Fields (Continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always 31.

Full Host Profile Record 5.1.1 Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.

Full Host Profile Record 5.1.1 Fields (Continued)

Field	Data Type	Description
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.

Full Host Profile Record 5.1.1 Fields (Continued)

Field	Data Type	Description
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — host • 1 — router • 2 — bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+ on page 331 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.

Full Host Profile Record 5.1.1 Fields (Continued)

Field	Data Type	Description
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/ VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.

Full Host Profile Record 5.1.1 Fields (Continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Sourcefire-detected IDs. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block on page 253 for a description of the data blocks in this list.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No

Full Host Profile Data Block 5.2.x

The Full Host Profile data block for version 5.2.x contains a full set of data describing one host. It has the format shown in the graphic below and explained in the following table. Note that, except for List data blocks, the graphic does not show the fields of the encapsulated data blocks. These encapsulated data blocks are described separately in Understanding Discovery & Connection Data Structures on page 164. The Full Host Profile data block a block type value of 140.

It supersedes the prior version, which has a block type of 135.

IMPORTANT! An asterisk (*) next to a block name in the following diagram indicates that multiple instances of the data block may occur.

	Byte 0	Byte 1	Byte 2	Byte 3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
	Full Host Profile Data Block (140)			
	Data Block Length			
	Host ID			
	Host ID, continued			
	Host ID, continued			
	Host ID, continued			
IP Addresses	List Block Type (11)			
	List Block Length			
	IP Address Data Blocks (143)*			
	Hops	Generic List Block Type (31)		
	Generic List Block Type, continued	Generic List Block Length		
OS Derived Fingerprints	Generic List Block Length, continued	Operating System Fingerprint Block Type (130)*		
	OS Fingerprint Block Type (130)*, con't	Operating System Fingerprint Block Length		
	OS Fingerprint Block Length, con't	Operating System Derived Fingerprint Data...		
	Generic List Block Type (31)			
	Generic List Block Length			
Server Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System Server Fingerprint Data...			
	Generic List Block Type (31)			

	Byte	0	1	2	3
	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
		Generic List Block Length			
Client Fingerprints		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System Client Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			
VDB Native Fingerprints 1		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System VDB Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			
VDB Native Fingerprints 2		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System VDB Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			
User Fingerprints		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System User Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			
Scan Fingerprints		Operating System Fingerprint Block Type (130)*			
		Operating System Fingerprint Block Length			
		Operating System Scan Fingerprint Data...			
		Generic List Block Type (31)			
		Generic List Block Length			

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td rowspan="3">Application Fingerprints</td><td colspan="32">Operating System Fingerprint Block Type (130)*</td></tr>
<tr><td colspan="32">Operating System Fingerprint Block Length</td></tr>
<tr><td colspan="32">Operating System Application Fingerprint Data...</td></tr>
<tr><td></td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td></td><td colspan="32">Generic List Block Length</td></tr>
<tr><td rowspan="3">Conflict Fingerprints</td><td colspan="32">Operating System Fingerprint Block Type (130)*</td></tr>
<tr><td colspan="32">Operating System Fingerprint Block Length</td></tr>
<tr><td colspan="32">Operating System Conflict Fingerprint Data...</td></tr>
<tr><td></td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td></td><td colspan="32">Generic List Block Length</td></tr>
<tr><td rowspan="3">Mobile Fingerprints</td><td colspan="32">Operating System Fingerprint Block Type (130)*</td></tr>
<tr><td colspan="32">Operating System Fingerprint Block Length</td></tr>
<tr><td colspan="32">Operating System Mobile Fingerprint Data...</td></tr>
<tr><td></td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td></td><td colspan="32">Generic List Block Length</td></tr>
<tr><td rowspan="3">IPv6 Server Fingerprints</td><td colspan="32">Operating System Fingerprint Block Type (130)*</td></tr>
<tr><td colspan="32">Operating System Fingerprint Block Length</td></tr>
<tr><td colspan="32">Operating System IPv6 Server Fingerprint Data...</td></tr>
<tr><td></td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td></td><td colspan="32">Generic List Block Length</td></tr>
<tr><td rowspan="3">Ipv6 Client Fingerprints</td><td colspan="32">Operating System Fingerprint Block Type (130)*</td></tr>
<tr><td colspan="32">Operating System Fingerprint Block Length</td></tr>
<tr><td colspan="32">Operating System Ipv6 Client Fingerprint Data...</td></tr>
<tr><td></td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td></td><td colspan="32">Generic List Block Length</td></tr>
</table>

Byte	0	1	2	3
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31
Ipv6 DHCP Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System IPv6 DHCP Fingerprint Data...			
	Generic List Block Type (31)			
	Generic List Block Length			
User Agent Fingerprints	Operating System Fingerprint Block Type (130)*			
	Operating System Fingerprint Block Length			
	Operating System User Agent Fingerprint Data...			
(TCP) Full Server Data	List Block Type (11)...			
	List Block Length...			
	(TCP) Full Server Data Blocks (104)*			
(UDP) Full Server Data	List Block Type (11)			
	List Block Length			
	(UDP) Full Server Data Blocks (104)*			
Network Protocol Data	List Block Type (11)			
	List Block Length			
	(Network) Protocol Data Blocks (4)*			
Transport Protocol Data	List Block Type (11)			
	List Block Length			
	(Transport) Protocol Data Blocks (4)*			
MAC Address Data	List Block Type (11)			
	List Block Length			
	Host MAC Address Data Blocks (95)*			
	Last Seen			
	Host Type			
	Business Criticality		VLAN ID	

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td></td><td colspan="8">VLAN Type</td><td colspan="8">VLAN Priority</td><td colspan="16">Generic List Block Type (31)</td></tr>
<tr><td rowspan="2">Host Client Data</td><td colspan="16">Generic List Block Type, continued</td><td colspan="16">Generic List Block Length</td></tr>
<tr><td colspan="16">Generic List Block Length, continued</td><td colspan="16">Full Host Client Application Data Blocks (112)*</td></tr>
<tr><td rowspan="3">NetBios Name</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">NetBIOS Name String...</td></tr>
<tr><td rowspan="3">Notes Data</td><td colspan="32">String Block Type (0)</td></tr>
<tr><td colspan="32">String Block Length</td></tr>
<tr><td colspan="32">Notes String....</td></tr>
<tr><td rowspan="3">(VDB) Host Vulns</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">(VDB) Host Vulnerability Data Blocks (85)*</td></tr>
<tr><td rowspan="3">3rd Pty/VDB) Host Vulns</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">(Third Party/VDB) Host Vulnerability Data Blocks (85)*</td></tr>
<tr><td rowspan="3">3rd Pty Scan Host Vulns</td><td colspan="32">Generic List Block Type (31)</td></tr>
<tr><td colspan="32">Generic List Block Length</td></tr>
<tr><td colspan="32">(Third Party Scan) Host Vulnerability Data Blocks with Original Vuln IDs (85)*</td></tr>
<tr><td rowspan="3">Attribute Value Data</td><td colspan="32">List Block Type (11)</td></tr>
<tr><td colspan="32">List Block Length</td></tr>
<tr><td colspan="32">Attribute Value Data Blocks *</td></tr>
<tr><td></td><td colspan="8">Mobile</td><td colspan="8">Jailbroken</td><td colspan="16"></td></tr>
</table>

The Full Host Profile Record 5.2.x Fields table describes the components of the Full Host Profile for 5.2.x record.

Full Host Profile Record 5.2.x Fields

FIELD	DATA TYPE	DESCRIPTION
Host ID	uint8[16]	Unique ID number of the host. This is a UUID.
List Block Type	uint32	Initiates a List data block comprising IP address data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated IP address data blocks.
IP Address	variable	IP addresses of the host and when each IP address was last seen. See Host IP Address Data Block on page 273 for a description of this data block.
Hops	uint8	Number of network hops from the host to the device.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data derived from the existing fingerprints for the host. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Derived Fingerprint Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host derived from the existing fingerprints for the host. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 1) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a Sourcefire VDB fingerprint. This value is always 31.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (VDB) Native Fingerprint 2) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using the fingerprints in the Sourcefire vulnerability database (VDB). See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a user. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a user. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by a vulnerability scanner. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Scan Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by a vulnerability scanner. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data added by an application. This value is always 31.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Application Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host added by an application. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data selected through fingerprint conflict resolution. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Conflict Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host selected through fingerprint conflict resolution. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying mobile device fingerprint data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Mobile) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a mobile device host. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 server fingerprint. This value is always 31.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an IPv6 DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (IPv6 DHCP) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an IPv6 DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a user agent fingerprint. This value is always 31.

Full Host Profile Record 5.2.x Fields (Continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (User Agent) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a user agent fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying TCP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(TCP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the TCP services on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Full Server data blocks conveying UDP service data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Full Server data blocks.
(UDP) Full Server Data Blocks *	variable	List of Full Server data blocks conveying data about the UDP sub-servers on the host. See Full Host Server Data Block 4.10.0+ on page 314 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
(Network) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the network protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus the length of all encapsulated Protocol data blocks.
(Transport) Protocol Data Blocks *	variable	List of Protocol data blocks conveying data about the transport protocols on the host. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block containing Host MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated Host MAC Address data blocks.
Host MAC Address Data Blocks *	variable	List of Host MAC Address data blocks. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates host type. Values include: • 0 — host • 1 — router • 2 — bridge • 3 — NAT (network address translation device) • 4 — LB (load balancer)
Business Criticality	uint16	Indicates criticality of host to business.
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying Client Application data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Client Application data blocks.
Full Host Client Application Data Blocks *	variable	List of Client Application data blocks. See Full Host Client Application Data Block 5.0+ on page 331 for a description of this data block.
String Block Type	uint32	Initiates a String data block for the host NetBIOS name. This value is always 0.
String Block Length	uint32	Number of bytes in the String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the NetBIOS name string.
NetBIOS Name	string	Host NetBIOS name string.
String Block Type	uint32	Initiates a String data block for host notes. This value is always 0.
String Block Length	uint32	Number of bytes in the notes String data block, including eight bytes for the string block type and length fields, plus the number of bytes in the notes string.
Notes	string	Contains the contents of the Notes host attribute for the host.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying VDB vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.

Full Host Profile Record 5.2.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
(VDB) Host Vulnerability Data Blocks *	variable	List of Host Vulnerability data blocks for vulnerabilities identified in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third-party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party/VDB) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner and containing information about host vulnerabilities cataloged in the Sourcefire vulnerability database (VDB). See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Host Vulnerability data blocks conveying third party scan vulnerability data. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated data blocks.
(Third Party Scan) Host Vulnerability Data Blocks *	variable	Host Vulnerability data blocks sourced from a third party scanner. Note that the host vulnerability IDs for these data blocks are the third party scanner IDs, not Sourcefire-detected IDs. See Host Vulnerability Data Block 4.9.0+ on page 293 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Attribute Value data blocks conveying attribute data. This value is always 11.
List Block Length	uint32	Number of bytes in the List data block, including the list header and all encapsulated data blocks.

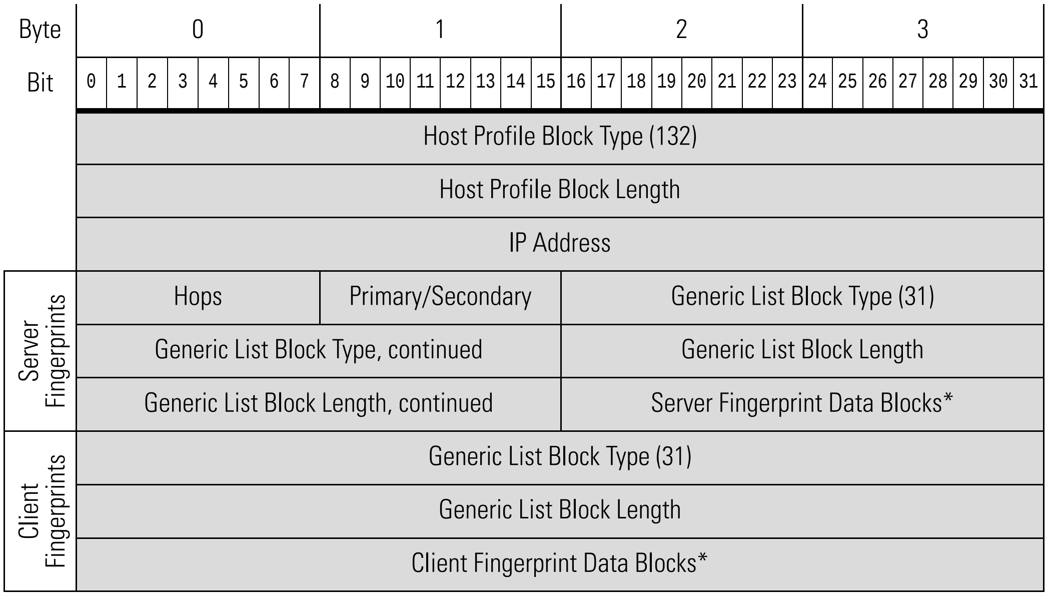
Full Host Profile Record 5.2.x Fields (Continued)

Field	Data Type	Description
Attribute Value Data Blocks *	variable	List of Attribute Value data blocks. See Attribute Value Data Block on page 253 for a description of the data blocks in this list.
Mobile	uint8	A true-false flag indicating whether the operating system is running on a mobile device.
Jailbroken	uint8	A true-false flag indicating whether the mobile device operating system is jailbroken.

Host Profile Data Block for 5.1.x

The following diagram shows the format of a Host Profile data block. The data block also does not include a host criticality value, but does include a VLAN presence indicator. In addition, a data block can convey a NetBIOS name for the host. The Host Profile data block has a block type of 132.

IMPORTANT! An asterisk(*) next to a block type field in the following diagram indicates the message may contain zero or more instances of the series 1 data block.


Byte
0
1
2
3
Bit
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Host Profile Block Type (132)
Host Profile Block Length
IP Address
Server Fingerprints
Hops
Primary/Secondary
Generic List Block Type (31)
Generic List Block Type, continued
Generic List Block Length
Generic List Block Length, continued
Server Fingerprint Data Blocks*
Client Fingerprints
Generic List Block Type (31)
Generic List Block Length
Client Fingerprint Data Blocks*

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
SMB Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	SMB Fingerprint Data Blocks*				
DHCP Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	DHCP Fingerprint Data Blocks*				
Mobile Device Fingerprints	Generic List Block Type (31)				
	Generic List Block Length				
	Mobile Device Fingerprint Data Blocks*				
TCP Server Block*	List Block Type (11)				List of TCP Servers
	List Block Length				
	TCP Server Data Blocks				
UDP Server Block*	List Block Type (11)				List of UDP Servers
	List Block Length				
	UDP Server Data Blocks				
Network Protocol Block*	List Block Type (11)				List of Network Protocols
	List Block Length				
	Network Protocol Data Blocks				
Transport Protocol Block*	List Block Type (11)				List of Transport Protocols
	List Block Length				
	Transport Protocol Data Blocks				
MAC Address Block*	List Block Type (11)				List of MAC Addresses
	List Block Length				
	Host MAC Address Data Blocks				
	Host Last Seen				
	Host Type				

Byte	0	1	2	3	
Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	
	Mobile	Jailbroken	VLAN Presence	VLAN ID	
Client App Data	VLAN ID, cont.	VLAN Type	VLAN Priority	Generic List Block Type (31)	List of Client Applications
	Generic List Block Type (31), cont.			Generic List Block Length	
	Generic List Block Length, cont.			Client Application Data Blocks	
NetBIOS Name	String Block Type (0)				
	String Block Length				
	NetBIOS String Data...				

The Host Profile Data Block 5.1.x Fields table describes the fields of the host profile data block returned by version 5.1.x

Host Profile Data Block 5.1.x Fields

FIELD	DATA TYPE	DESCRIPTION
Host Profile Block Type	uint32	Initiates the Host Profile data block for 5.1.x. This value is always 132.
Host Profile Block Length	uint32	Number of bytes in the Host Profile data block, including eight bytes for the host profile block type and length fields, plus the number of bytes included in the host profile data that follows.
IP Address	uint8[4]	IP address of the host described in the profile, in IP address octets.
Hops	uint8	Number of hops from the host to the device.
Primary/ Secondary	uint8	Indicates whether the host is in the primary or secondary network of the device that detected it: • 0 — host is in the primary network. • 1 — host is in the secondary network.

Host Profile Data Block 5.1.x Fields (Continued)

Field	Data Type	Description
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a server fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Server Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a server fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a client fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Client Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a client fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using an SMB fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (SMB Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using an SMB fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.

Host Profile Data Block 5.1.x Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (DHCP Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a DHCP fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Operating System Fingerprint data blocks conveying fingerprint data identified using a DHCP fingerprint. This value is always 31.
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated Operating System Fingerprint data blocks.
Operating System Fingerprint (Mobile Device Fingerprint) Data Blocks *	variable	Operating System Fingerprint data blocks containing information about the operating system on a host identified using a mobile device fingerprint. See Operating System Fingerprint Data Block 5.1+ on page 339 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying TCP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
TCP Server Data Blocks	variable	Host server data blocks describing a TCP server. See Host Server Data Block for Version 4.9.0.x on page 516 for a description of this data block.

Host Profile Data Block 5.1.x Fields (Continued)

Field	Data Type	Description
List Block Type	uint32	Initiates a List data block comprising Server data blocks conveying UDP server data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Server data blocks. This field is followed by zero or more Server data blocks.
UDP Server Data Blocks	uint32	Host server data blocks describing a UDP server. See Host Server Data Block for Version 4.9.0.x on page 516 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying network protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more Protocol data blocks.
Network Protocol Data Blocks	uint32	Protocol data blocks describing a network protocol. See Protocol Data Block on page 243 for a description of this data block.
List Block Type	uint32	Initiates a List data block comprising Protocol data blocks conveying transport protocol data. This value is always 11.
List Block Length	uint32	Number of bytes in the list. This number includes the eight bytes of the list block type and length fields, plus all encapsulated Protocol data blocks. This field is followed by zero or more transport protocol data blocks.
Transport Protocol Data Blocks	uint32	Protocol data blocks describing a transport protocol. See Protocol Data Block on page 243 for a description of this data block.

Host Profile Data Block 5.1.x Fields (Continued)

Field	Data Type	Description
List Block Type	uint32	Initiates a List data block comprising MAC Address data blocks. This value is always 11.
List Block Length	uint32	Number of bytes in the list, including the list header and all encapsulated MAC Address data blocks.
Host MAC Address Data Blocks	uint32	Host MAC Address data blocks describing a host MAC address. See Host MAC Address 4.9+ on page 297 for a description of this data block.
Host Last Seen	uint32	UNIX timestamp that represents the last time the system detected host activity.
Host Type	uint32	Indicates the host type. The following values may appear: • 0 — host • 1 — router • 2 — bridge • 3 — NAT device • 4 — LB (load balancer)
Mobile	uint8	True-false flag indicating whether the host is a mobile device.
Jailbroken	uint8	True-false flag indicating whether the host is a mobile device that is also jailbroken.
VLAN Presence	uint8	Indicates whether a VLAN is present: • 0 — Yes • 1 — No
VLAN ID	uint16	VLAN identification number that indicates which VLAN the host is a member of.
VLAN Type	uint8	Type of packet encapsulated in the VLAN tag.
VLAN Priority	uint8	Priority value included in the VLAN tag.
Generic List Block Type	uint32	Initiates a Generic List data block comprising Client Application data blocks conveying client application data. This value is always 31.

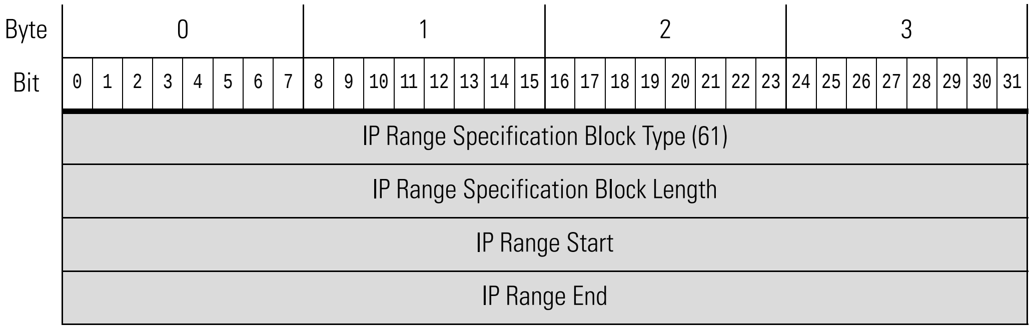
Host Profile Data Block 5.1.x Fields (Continued)

Field	Data Type	Description
Generic List Block Length	uint32	Number of bytes in the Generic List data block, including the list header and all encapsulated client application data blocks.
Client Application Data Blocks	uint32	Client application data blocks describing a client application. See Full Host Client Application Data Block 5.0+ on page 331 for a description of this data block.
String Block Type	uint32	Initiates a string data block for the NetBIOS name. This value is set to 0 to indicate string data.
String Block Length	uint32	Indicates the number of bytes in the NetBIOS name data block, including eight bytes for the string block type and length, plus the number of bytes in the NetBIOS name.
NetBIOS String Data	Variable	Contains the NetBIOS name of the host described in the host profile.

IP Range Specification Data Block for 4.7.x - 5.1.1.x

The IP Range Specification data block conveys a range of IP addresses. IP Range Specification data blocks are used in User Protocol, User Client Application, Address Specification, User Product, User Server, User Hosts, User Vulnerability, User Criticality, and User Attribute Value data blocks. The IP Range Specification data block has a block type of 61.

The following diagram shows the format of the IP Range Specification data block:



The IP Range Specification Data Block Fields table describes the components of the IP Range Specification data block.

IP Range Specification Data Block Fields

Field	Data Type	Description
IP Range Specification Block Type	uint32	Initiates a IP Range Specification data block. This value is always 61.
IP Range Specification Block Length	uint32	Total number of bytes in the IP Range Specification data block, including eight bytes for the IP Range Specification block type and length fields, plus the number of bytes of IP range specification data that follows.
IP Range Specification Start	uint32	The starting IP address for the IP address range.
IP Range Specification End	uint32	The ending IP address for the IP address range.

Legacy Metadata Structures

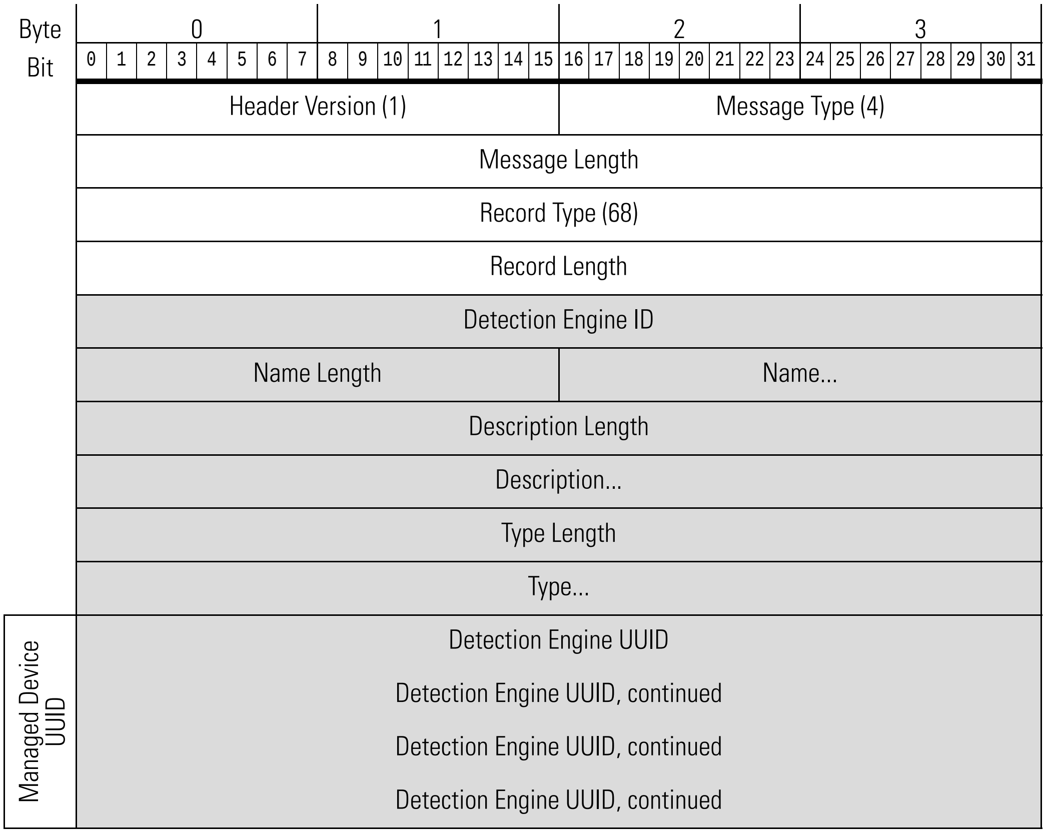
The following legacy data structures apply to versions of the system before 5.1:

Detection Engine Record for 4.6.1 - 4.10.x on page 719

Detection Engine Record for 4.6.1 - 4.10.x

The eStreamer service transmits metadata containing device information for an event within a Detection Engine record, the format of which is shown below.

The Detection Engine for 4.6.1+ contains the same fields as the Detection Engine record for 4.6 but has a new UUID field. Detection Engine information is sent when the Version 3 or Version 4 metadata flag—bit 15 or bit 20 in the Request Flags field of a request message—is set. See Request Flags on page 30. The Record Type field has a value of 68.



The Detection Engine Record Fields table describes the fields in the Detection Engine Record.

Detection Engine Record Fields

FIELD	DATA TYPE	DESCRIPTION
Detection Engine ID	uint32	The detection engine ID number.
Name Length	uint16	The number of bytes included in the detection engine name.
Name	string	The name of the detection engine that detected the event.
Description Length	uint16	The number of bytes included in the detection engine description.

Detection Engine Record Fields (Continued)

FIELD	DATA TYPE	DESCRIPTION
Description	string	The description of the detection engine that detected the event.
Type Length	uint16	The number of bytes included in the detection engine type.
Type	string	The type of the detection engine that detected the event.
UUID	uint8[16]	A detection engine ID number that acts as a unique identifier for the detection engine.

Index