Stealthwatch System APIs

(for Stealthwatch System v6.9)

Copyrights and Trademarks

NOTICE

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Contents

1

About the Stealthwatch System APIs

Overview

The Stealthwatch System REST API consists of a collection of resources for developers, administrators, or partners that enable the functionality of Stealthwatch to be accessed programmatically. Since the Stealthwatch REST API is based on open standards, you can use any programming or scripting language you wish as long as it supports HTTP.

This document assumes you have some knowledge of using REST services and a basic understanding of the Hyper Text Transfer Protocol (HTTP). Experience using either curl or wget is highly beneficial. To access the REST API resource reference from a running SMC, go to https://<smcip>/smc/restapi-docs/index.html.

You may also be interested in the "SMC Web Services Programming Guide" which documents the SOAP interface to Stealthwatch. With the Web Services API that is compliant with the Simple Object Access Protocol (SOAP), administrators can use external applications, such as Security Information and Event Management (SIEM) systems, trouble-ticketing systems, and third-party reporting systems, to access data from the SMC. For more information, refer to the SMC Web Services Programming Guide.

The Stealthwatch Reporting Service provides the following APIs:

- Authentication
- Policy Service
- Reporting
- Domain

Using the Stealthwatch REST API

Authentication

Before you can use the Stealthwatch REST API, you need to authenticate. The same credentials (login/password pair) you use to login to the user interface for Stealthwatch can be used for accessing the Stealthwatch REST API. If you do not have credentials, the first step is to contact your Stealthwatch administrator.

You authenticate by sending a POST request containing the password to Stealthwatch. Assuming your Stealthwatch Management Console (SMC) is at "smcaddress", the username is "jim" and the password is "password123", an example of using curl to authenticate is shown below:

```
curl -s -k -c cookies.txt -d
"username=jim&password=password123"
https://smcaddress/token/v2/authenticate
```

Assuming the credentials are good, a user session is created and a cookie is returned in the file cookies.txt. You will need to reference the cookie in subsequent calls. Note that the user session will expire after a period of inactivity - this is no different to user sessions initiated through logging in via the browser.

HTTP Methods

The standard HTTP methods are used to access the Stealthwatch System REST API. The verbs are applied consistently across all resources and should be fairly familiar to any other REST APIs you may have used:

Desired Option	HTTP Verb to Use
Read	GET
Create	POST
Update	PUT
Delete	DELETE

Please note that not all resources support all HTTP methods. For example, some resources may only support GET.

HTTP Status Codes

If the operation succeeded, expect a standard 200 OK response. You should always check the Status-Code field in the HTTP header. If it is anything other than 200, then the operation failed. The error codes are in the following table:

HTTP Status Code	Description
400 Bad Request	General error when fulfilling the request that would cause an invalid state, for example, domain validation errors or missing data.
401 Unauthorized	Error code response for missing or invalid authentication token. It indicates that the request can be retried once a valid authentication token has been acquired.

404 Not Found	Used when the requested resource is not found, whether because it does not exist, or for security reasons, the service wants to mask its existence.
405 Method Not Allowed	A request was made of a resource using a request method not supported by that resource, for example, using GET on a write-only resource or using PUT on a read-only resource.
500 Internal Server Error	The general catch-all error when the server-side has an exception. In the event of a failure, the response body will contain a JSON response that should contain more information about what caused the operation to fail.

Media Types

The Stealthwatch REST API returns HTTP responses in JSON format only.

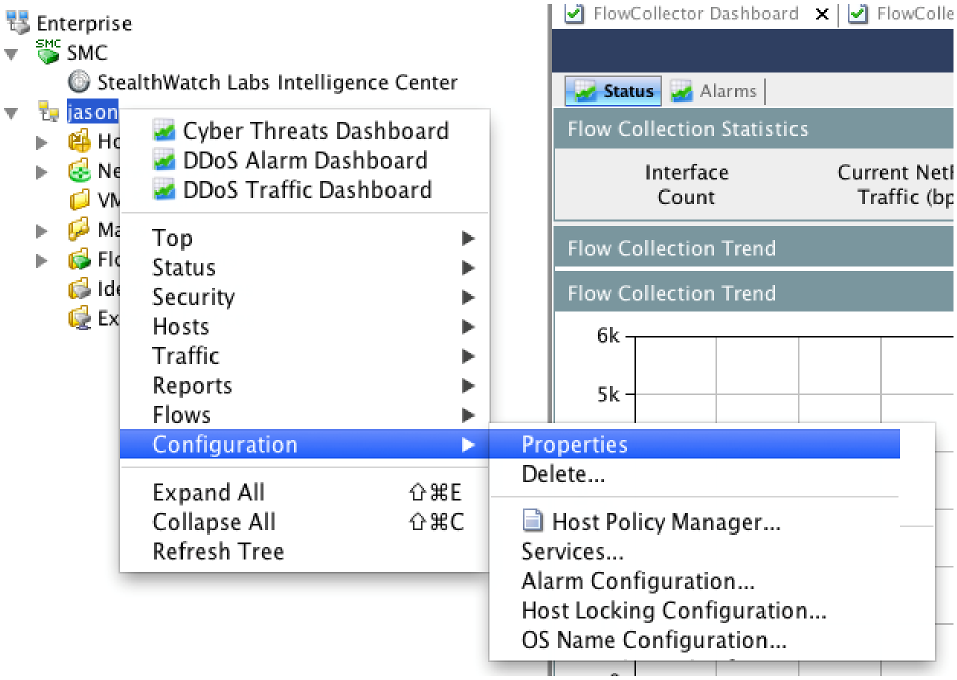
Parameters

Several of the REST API calls may require identifiers of various objects in the system, e.g. domainId. This section describes how to obtain these identifiers by two methods.

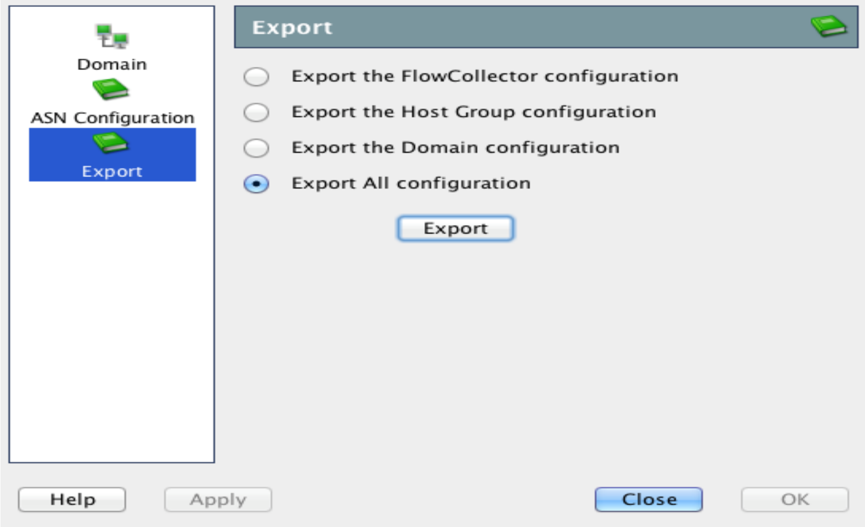
SMC Client

One way to obtain these identifiers is via the Stealthwatch Management Console (SMC) client as follows:

1. In the SMC client interface, elect your domain in the enterprise tree and then click **Configuration > Properties**.



The Properties dialog opens.



○ Export the FlowCollector configuration
○ Export the Host Group configuration
○ Export the Domain configuration
● Export All configuration

2. Click **Export** and select the "Export All configuration” option.
3. Save the XML configuration file.
4. After it is downloaded, open it with a text editor.
5. Locate the domainId by search for “<domain id”.
6. Locate the hostGroupId by searching for “<host-group”.

7. Locate the interface if-index by searching for “<interface if-index=”.
8. Locate the exporterIp by searching for “<exporter ip=".

Command Line Interface

You can also acquire parameter information from a CLI. Type this command to get a list of the host_id from a Flow Collector:

```
grep id= /lancope/var/sw/today/config/groups.xml | awk '
{print $2, $3, $4}' | sed s/\"//g| sed s/id=//g |awk
'$1<60000'|sort -k1,1n |less
```

To get the domain number for an SMC, type this command:

```
ls /lancope/var/smc/config/ | grep domain
```

2

AUTHENTICATION APIS

The authentication APIs include the following:

- Tokens API
  - Provides access to
    - communicate with the Stealthwatch System using authentication end points.
    - validate, renew, and clear tokens.

Delete Token

This authentication API clears the token cookie:

- DELETE /token

Response

HTTP status code 204

The token has been set to an empty string.

Headers

- Set-Cookie: *(string)*

  The exact cookie named is configurable, but the default is named jwt. This cookie is restricted to not be available to javascript in a browser, and to be transmitted only over https.

Example: jwt=""; Secure; HttpOnly; Path=/

Get Token

This authentication API validates the provided token (generally used only internally):

- GET /token

Request

Headers

- Cookie: *required (string)*

The JWT, previously created in a cookie named based on the configuration.

Example: jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva G4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Response

HTTP status code 204

The token provided was valid.

HTTP status code 401

Provided credentials are not valid, passphrase has expired, user is disabled, etc.

Headers

- WWW-Authenticate: *(string - pattern: token-authority)*
  Identifies what your authentication should be used for.

Example: token-authority

Post Token

This authentication API renews a valid JWT token:

- POST /token

Request

Headers

- Cookie: *required (string)*
  The JWT, previously created in a cookie named based on the configuration.

Example: jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva G4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Response

HTTP status code 204

The token provided was valid and renewable.

Headers

- Set-Cookie: *(string)*

The JWT is returned in a cookie. The exact cookie named is configurable, but the default is named jwt. This cookie is restricted to not be available to javascript in a browser, and to be transmitted only over https.

Example:
jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ;
Secure; HttpOnly; Path=/

HTTP status code 401

The token as not valid or is no longer renewable.

Headers

- WWW-Authenticate: *(string - pattern: token-authority)*

  Identifies what your authentication should be used for.

Example: token-authority

Post Token v2

This authentication API attempts to obtain a token for the provided username:

- POST /token/v2

Request

Body

- Type: application/x-www-form-urlencoded; charset=UTF-8
- Form Parameters:
  - username: *required (string)*

    The identity of the user to authenticate.
  - password: *required (string)*

    The password or phrase used to authenticate.

Response

HTTP status code 200

Credentials approved.

Headers

- Set-Cookie: *(string)*

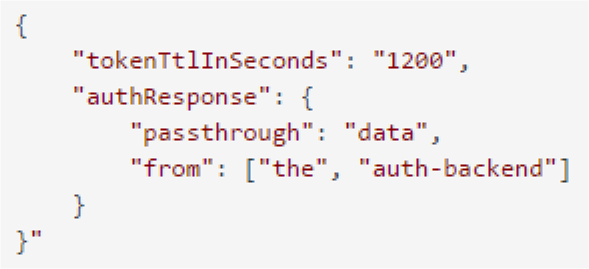
The JWT is returned in a cookie. The exact cookie named is configurable, but the default is named jwt. This cookie is restricted to not be available to javascript in a browser, and to be transmitted only over https.

Example:
jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ; Secure; HttpOnly; Path=/

Body

- Type: application/json

  Example



```
{
    "tokenTtlInSeconds": "1200",
    "authResponse": {
        "passthrough": "data",
        "from": ["the", "auth-backend"]
    }
}"
```

HTTP status code 401

Provided credentials are not valid, passphrase has expired, user is disabled, etc.

Headers

- WWW-Authenticate: *(string - pattern: token-authority)*

  Identifies what your authentication should be used for.

Example: token-authority

3

Domain APIs

The domain APIs include the following:

- Exporter API
  - Provides a list of time series data with inbound and outbound interface application traffic, by application.
- Host Group API
  - Provides a list of time series data
    - for the default host group.
    - with inbound, outbound and within application traffic.

Domain Exporter and Host Group APIs

The full documentation for all the available resources exposed through the Stealthwatch REST API is available at the following URL (where smcaddress should be replaced with the domain name or IP address of your specific SMC):

```
https://smcaddress/smc/restapi-docs/index.html
```

We built the resource reference using Swagger. This enables you not only to inspect the detailed specification for each resource, it also enables you to try out the Stealthwatch REST API from the browser without having to write a single line of code.

This API generates a list of time series data with inbound and outbound interface application traffic, by application:

- https://[serviceAddress/smc/rest/domains/{domainId}/exporters/{flowCollectorDeviceId}/{exporterIp}/{interface}/interfaceApplicationTraffic

Request

URI Parameters

- domainId: *required (integer)* The domain Id.
- flowCollectorDeviceId: *required (integer)* The device ID of the Flow Collector associated with the exporter.
- exporterIp: *required (integer)* The exporter IP address.
- interface: *required (integer)* The interface ID (ifindex) of the exporter to report on.

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- start: (integer)

  Optional start date/time in ISO 8601 format.
- end: (integer)

  Optional end date/time in ISO 8601 format.

Response

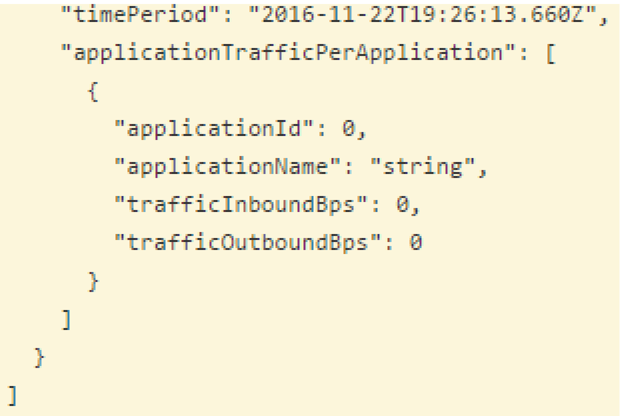
HTTP status code 200

???

Body

- Type: application/json

  Example



```
    "timePeriod": "2016-11-22T19:26:13.660Z",
    "applicationTrafficPerApplication": [
      {
        "applicationId": 0,
        "applicationName": "string",
        "trafficInboundBps": 0,
        "trafficOutboundBps": 0
      }
    ]
  }
]
```

This API generates a a list of time series data for the default host group. This API returns only the top ten application types. Application traffic lists will include zero value traffic statistics for application types missing in any given time segment.

- https://[serviceAddress/smc/rest/domains/{domainId}/hostgroups/dashboard

Note: Similar reports can be generated using the Host Group Application Traffic APIs.

Request

URI Parameters

- domainId: *required (integer)* The domain Id.

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- start: (integer)

  Optional start date/time in ISO 8601 format.
- end: (integer)

  Optional end date/time in ISO 8601 format.

Response

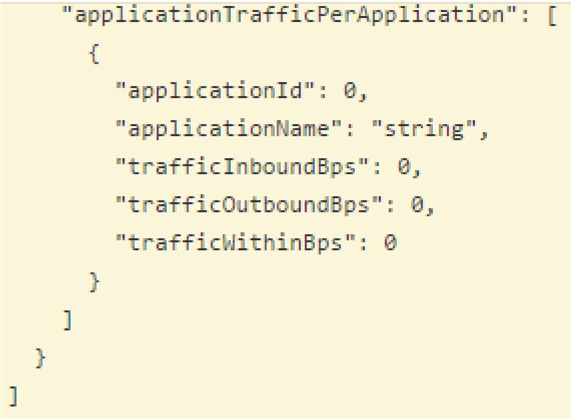
HTTP status code 200

???

Body

- Type: application/json

  Example



```
    "applicationTrafficPerApplication": [
      {
        "applicationId": 0,
        "applicationName": "string",
        "trafficInboundBps": 0,
        "trafficOutboundBps": 0,
        "trafficWithinBps": 0
      }
    ]
  }
]
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

This API generates a list of time series data with inbound, outbound and within application traffic:

- https://[serviceAddress/smc/rest/domains/{domainId}/hostgroups/{hostGroupId}/applicationTraffic

Note: Similar reports can be generated using the Host Group Application Traffic APIs.

Request

URI Parameters

- domainId: *required (integer)* The domain Id.
- host GroupId: *required (integer)* The host group Id.

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- start: (integer)

  Optional start date/time in ISO 8601 format.
- end: (integer)

  Optional end date/time in ISO 8601 format.

Response

HTTP status code 200

???

Body

- Type: application/json

  Example

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

Policy Service APIs

The policy service APIs include the following:

- Flow Filter API
    - Provides access to associated flow filters of a security event.

Filter Flow Security Event APIs

This policy service API retrieves a flow filter json for the given security event type id:

- https://[serviceAddress/sw-policy/v1/tenants/{tentantId}/filter/flow/securityEvents

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- typeId: *required (integer)*

  Security Event Type ID

Example: 32

- firstActiveTime: *required (date)*

  Security Event First Active Time. DateFormat should be in UTC.

Example: 2014-08-28T06:44:03Z

- lastActiveTime: *required (date)*

  Security Event Last Active Time. DateFormat should be in UTC.

Example: 2014-08-28T06:49:03Z

- sourceIp: *required (string)*
  Security Event Source IP

Example: 192.2.3.5

- targetIp: *required (string)*
  Security Event Target IP

Example: 192.2.4.3

- port: *(integer)*
  Security Event Port

Example: 80

Response

HTTP status code 200

OK.

Body

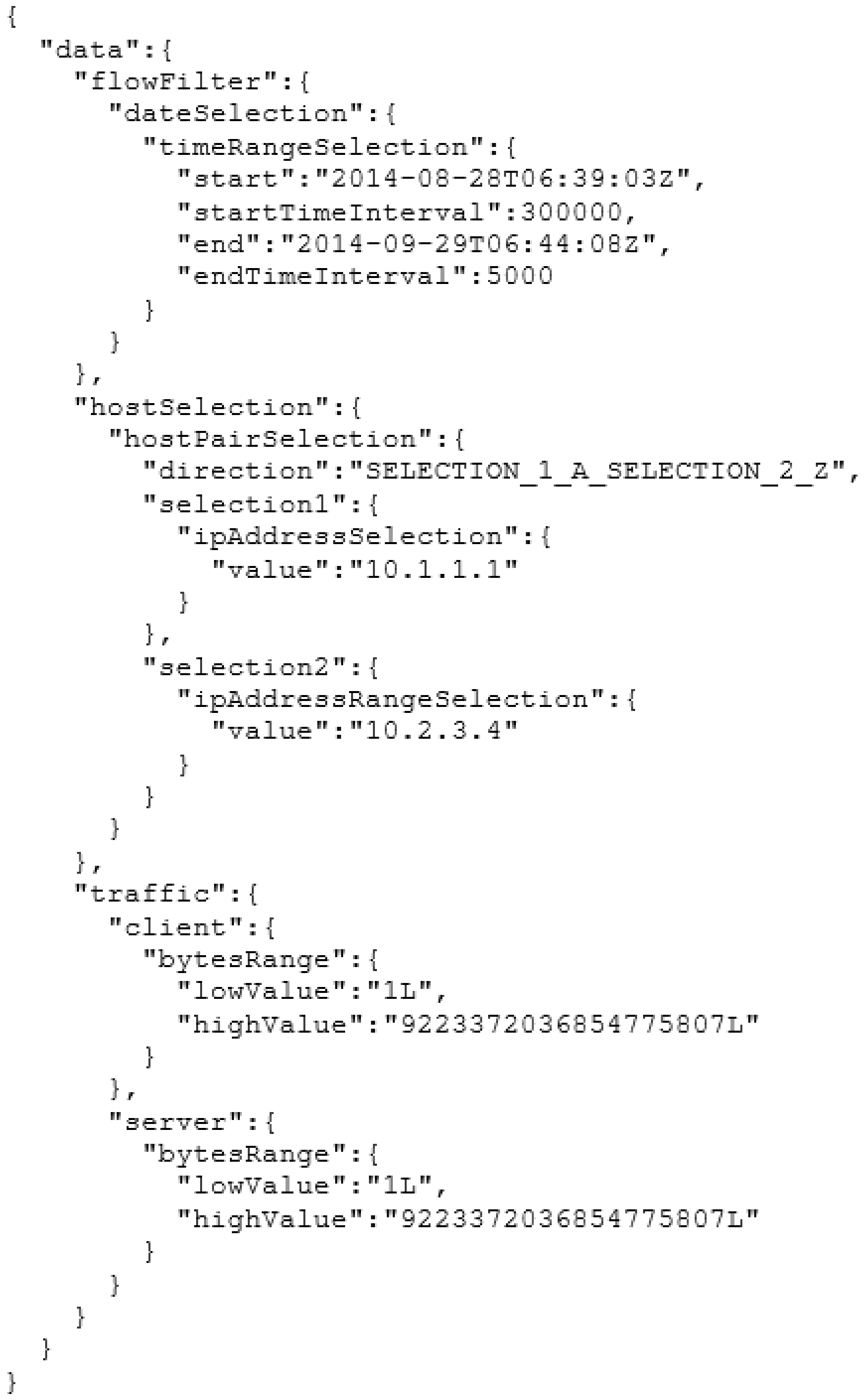
- Type: application/json

Example:
fjwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ;
Secure; HttpOnly; Path=/

Body

- Type: application/json

Example



```
{
  "data":{
    "flowFilter":{
      "dateSelection":{
        "timeRangeSelection":{
          "start":"2014-08-28T06:39:03Z",
          "startTimeInterval":300000,
          "end":"2014-09-29T06:44:08Z",
          "endTimeInterval":5000
        }
      }
    },
    "hostSelection":{
      "hostPairSelection":{
        "direction":"SELECTION_1_A_SELECTION_2_Z",
        "selection1":{
          "ipAddressSelection":{
            "value":"10.1.1.1"
          }
        },
        "selection2":{
          "ipAddressRangeSelection":{
            "value":"10.2.3.4"
          }
        }
      }
    },
    "traffic":{
      "client":{
        "bytesRange":{
          "lowValue":"1L",
          "highValue":"9223372036854775807L"
        }
      },
      "server":{
        "bytesRange":{
          "lowValue":"1L",
          "highValue":"9223372036854775807L"
        }
      }
    }
  }
}
```

HTTP status code 400

Missing token. Client should authenticate (Or) Invalid Query Parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

5

Reporting APIs

The reporting APIs include the following:

- Tenant and Tags API
  - Provides access to basic information about the Tenants (domains) and the Tags (host groups) in Stealthwatch. The results of the API are based on permissions. The API can fetch a list of tenants, and for each tenant it can get tags as a list or organized in a tree.
- Alarm Trends API
  - Provides access to daily alarm trend data. The API can fetch daily trend of an alarm type for up to 30 days. The trend aggregates the number of hosts (source and target) that have fired alarms for the specified type. The trend also calculates the maximum severity for the day, where severity is the ratio of alarm points accumulated toward the policy threshold.
- Host Group Application Traffic Reporting API
  - Provides access to hourly traffic trend data for tags and application(s) for a tag. The traffic trend reports on the number of bytes that are
    - sent from hosts classified by the tag.
    - received by the hosts classified by the tag.
    - transferred between (within) hosts classified by the tag for the hour.

    The API can fetch hourly traffic trend for a tag for up to 30 days.
- Top Alarming Hosts API
  - Provides access to top alarming hosts data. The API can fetch top alarming hosts for up to 30 days. The top alarming hosts are sorted according to the following criteria:
    - Primary sort is by hosts with alarms that have one or more contributing "always bad" security events (for both source and target).
    - Secondary sort is by hosts based on severity of alarm (for both source and target).

Tenant APIs

This API retrieves the list of tenants:

- https://[serviceAddress/sw-reporting/v1/tenants

Request

Headers

- Cookie: *required (string)*
  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

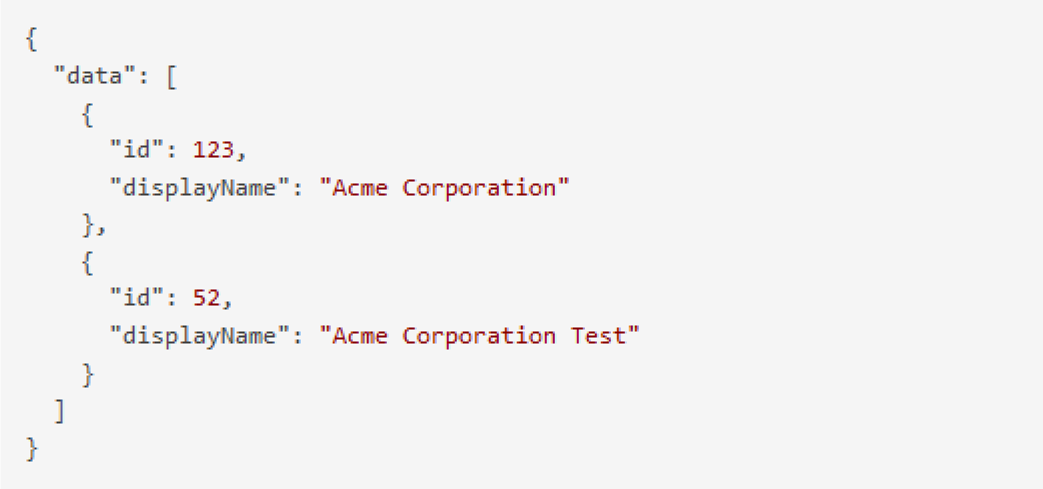
Response

HTTP status code 200

Body

- Type: application/json

Example



```
{
  "data": [
    {
      "id": 123,
      "displayName": "Acme Corporation"
    },
    {
      "id": 52,
      "displayName": "Acme Corporation Test"
    }
  ]
}
```

HTTP status code 400

Missing token. Client should authenticate.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

Retrieves a single tenant

This API retrieves a single tenant (that is given an ID):

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

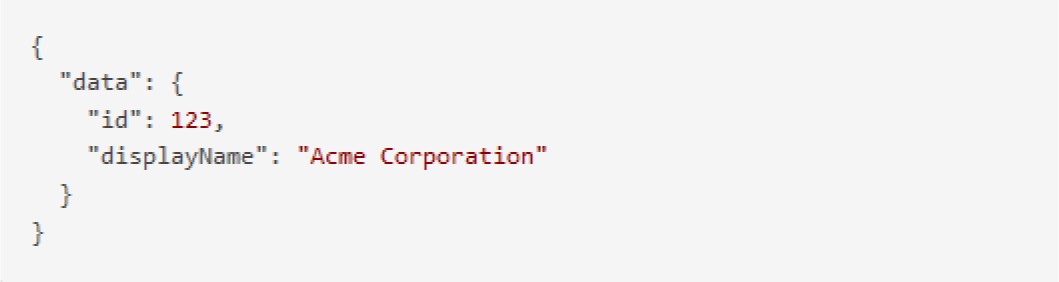
Response

HTTP status code 200

Body

- Type: application/json

  Example



```
{
  "data": {
    "id": 123,
    "displayName": "Acme Corporation"
  }
}
```

HTTP status code 400

Missing token. Client should authenticate.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

Tag APIs

These APIs retrieve all Tags in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

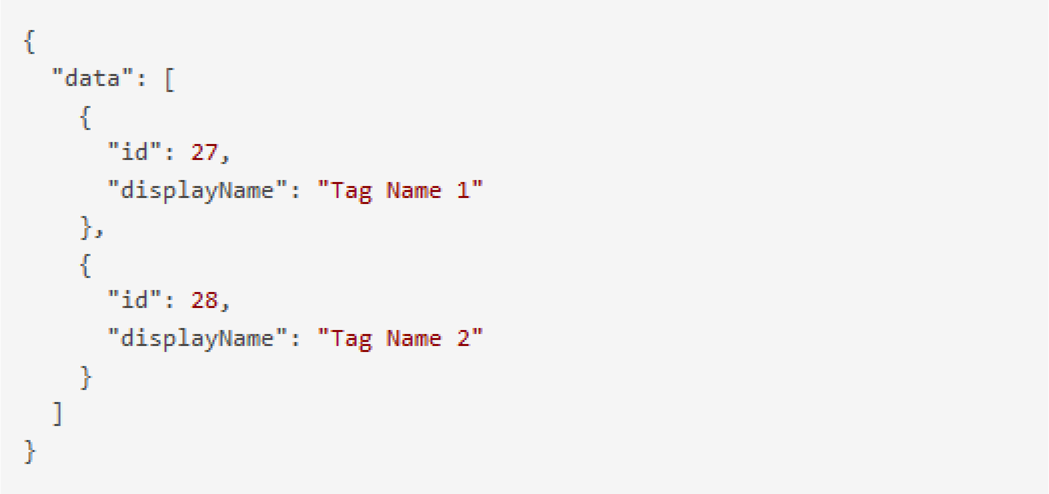
Response

HTTP status code 200

Body

- Type: application/json

  Example



```
{
  "data": [
    {
      "id": 27,
      "displayName": "Tag Name 1"
    },
    {
      "id": 28,
      "displayName": "Tag Name 2"
    }
  ]
}
```

HTTP status code 400

Missing token. Client should authenticate.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters

These APIs retrieve a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

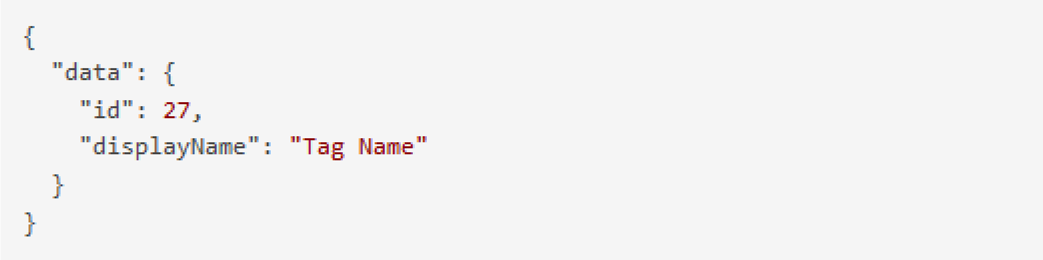
Response

HTTP status code 200

Body

- Type: application/json

  Example



```
{
  "data": {
    "id": 27,
    "displayName": "Tag Name"
  }
}
```

HTTP status code 400

Missing token. Client should authenticate.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters

These APIs retrieve all Tags in a Tenant organized in a hierarchy:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/tree
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/tree
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/tree
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/tree
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/tree

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*
  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

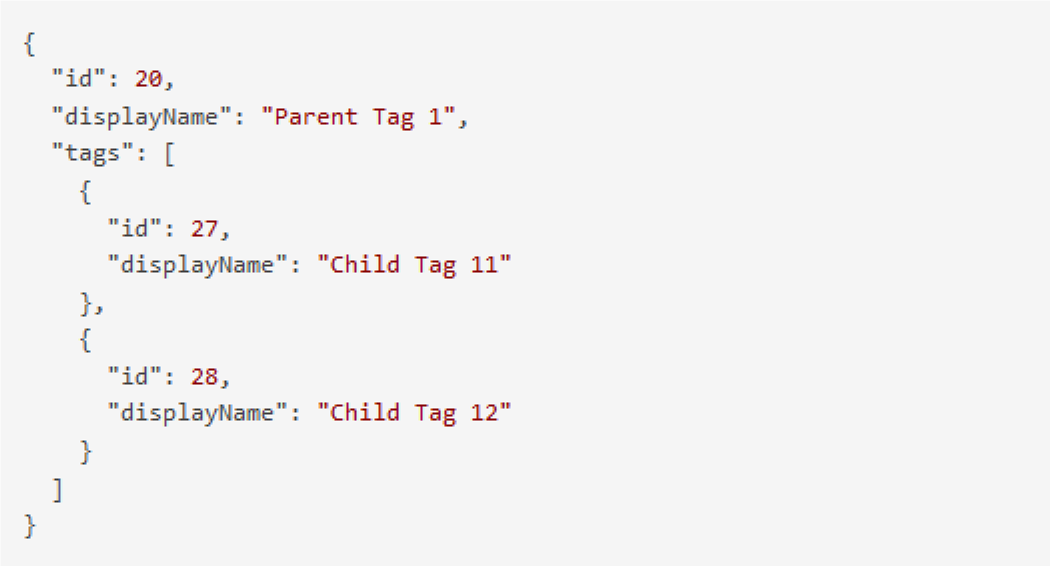
Response

HTTP status code 200

Body

- Type: application/json

Example



```
{
  "id": 20,
  "displayName": "Parent Tag 1",
  "tags": [
    {
      "id": 27,
      "displayName": "Child Tag 11"
    },
    {
      "id": 28,
      "displayName": "Child Tag 12"
    }
  ]
}
```

HTTP status code 400

Missing token. Client should authenticate.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

Alarm Trend APIs

These APIs retrieve the daily alarm trend for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/alarms/{alarmTypeId}/trend/daily

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*
- alarmTypeID: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is a 7 day alarm trend.

The response has a header element with the following fields:

- startTime: start time for the alarm trend.
- endTime: end time for the alarm trend.

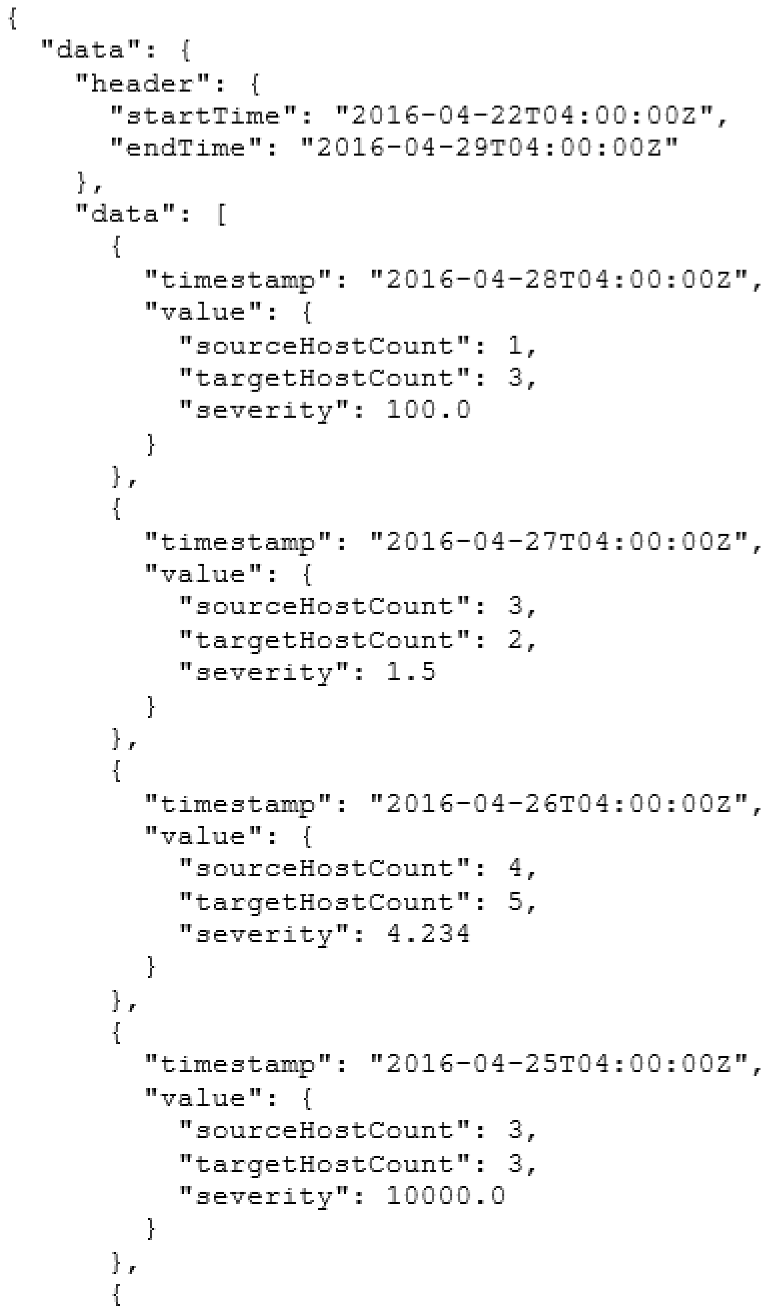
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For alarm trend this is the reset hour for the day.
- value: For alarm trends, this element has the following aggregated values.
  - sourceCount: Number of unique hosts that are source of alarms for the day. Only hosts that the user have access to are counted.
  - targetCount: Number of unique hosts that are target of alarms for the day. Only hosts that the user have access to are counted.
  - severity: This is the maximum severity of all the alarms for the day. Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

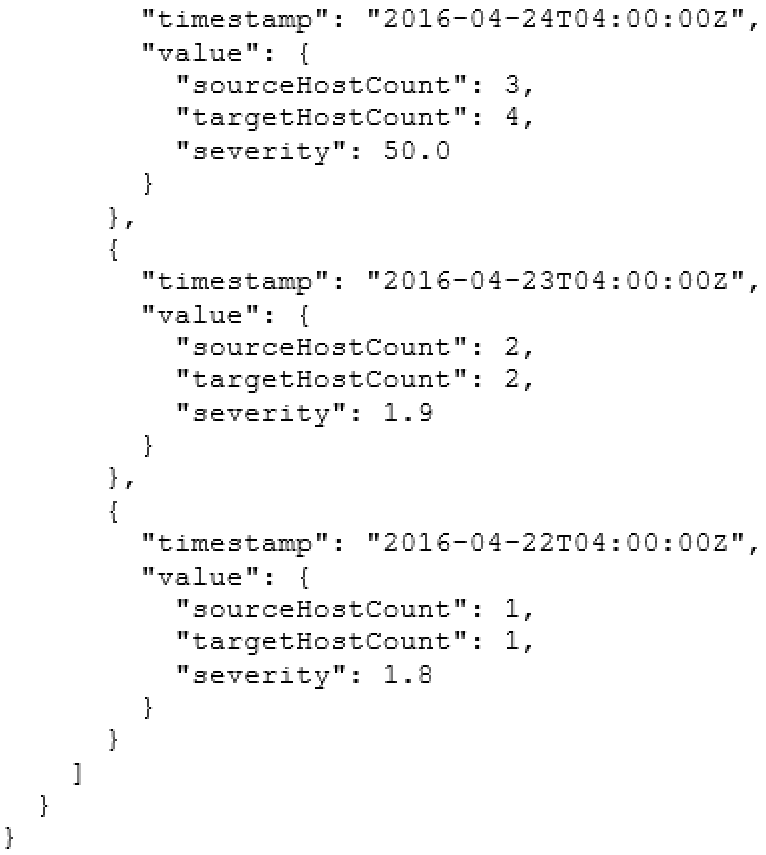
Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
```



```
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the daily alarm trend for a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/alarms/{alarmTypeId}/trend/daily
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/alarms/{alarmTypeId}/trend/daily

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is a 7 day alarm trend.

A trend has a header element with the following fields:

- startTime: start time for the alarm trend.
- endTime: end time for the alarm trend.

It has a data element which represents the time series data. Each element in the series has the following fields:

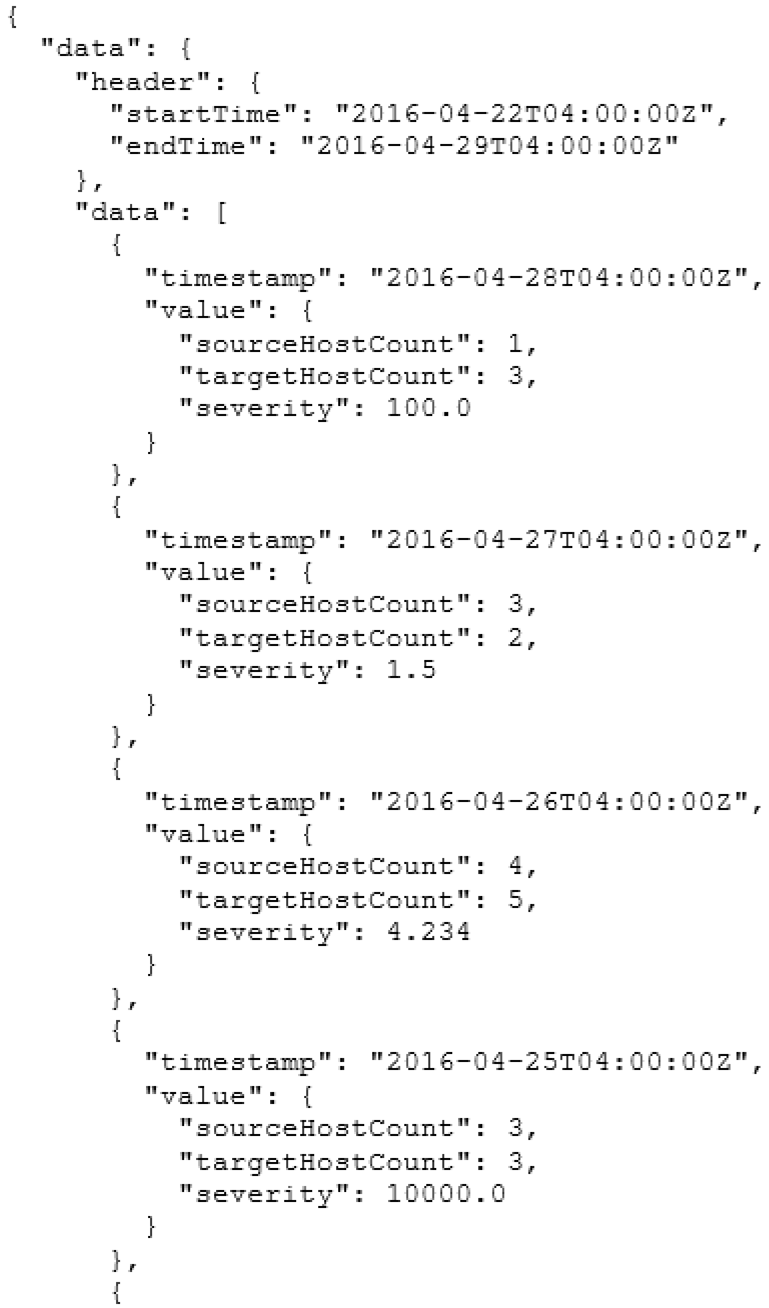
- timestamp: For alarm trend this is the reset hour for the day.
- value: For alarm trends, this element has the following aggregated values.

  - sourceCount: Number of unique hosts that are source of alarms for the day. Only hosts that the user have access to are counted.
  - targetCount: Number of unique hosts that are target of alarms for the day. Only hosts that the user have access to are counted.
  - severity: This is the maximum severity of all the alarms for the day. Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.

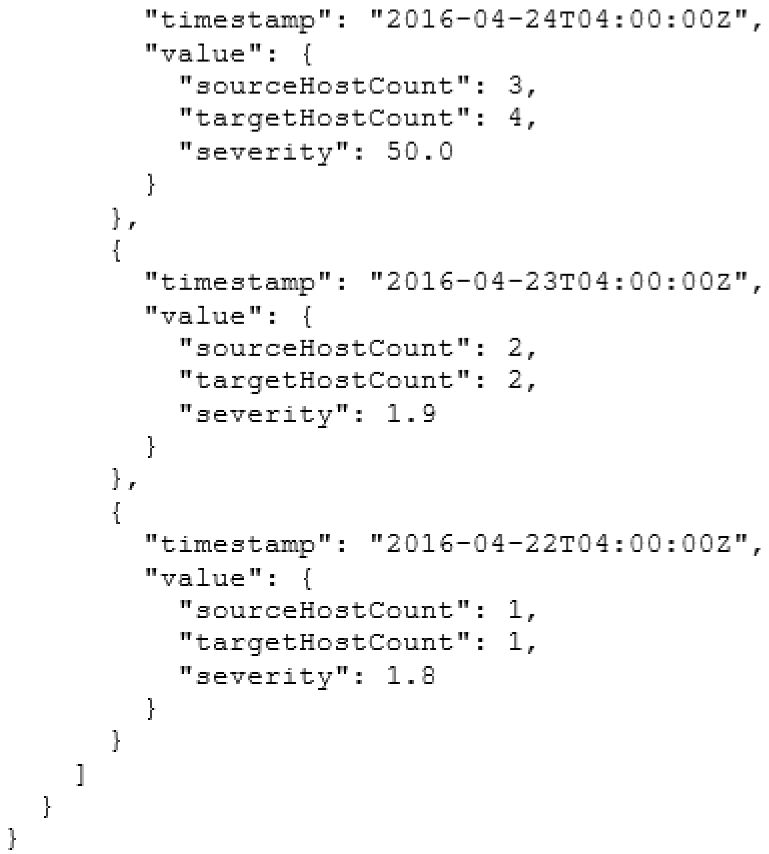
Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-22T04:00:00Z",
      "endTime": "2016-04-29T04:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 3,
          "severity": 100.0
        }
      },
      {
        "timestamp": "2016-04-27T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 2,
          "severity": 1.5
        }
      },
      {
        "timestamp": "2016-04-26T04:00:00Z",
        "value": {
          "sourceHostCount": 4,
          "targetHostCount": 5,
          "severity": 4.234
        }
      },
      {
        "timestamp": "2016-04-25T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 3,
          "severity": 10000.0
        }
      },
      {
```



```
        "timestamp": "2016-04-24T04:00:00Z",
        "value": {
          "sourceHostCount": 3,
          "targetHostCount": 4,
          "severity": 50.0
        }
      },
      {
        "timestamp": "2016-04-23T04:00:00Z",
        "value": {
          "sourceHostCount": 2,
          "targetHostCount": 2,
          "severity": 1.9
        }
      },
      {
        "timestamp": "2016-04-22T04:00:00Z",
        "value": {
          "sourceHostCount": 1,
          "targetHostCount": 1,
          "severity": 1.8
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

Host Group Application Traffic APIs

These APIs retrieve the hourly traffic trend of an application for a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/applications/{applicationId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/applications/{applicationId}/traffic/hourly

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/applications/{applicationId}/traffic/hourly

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

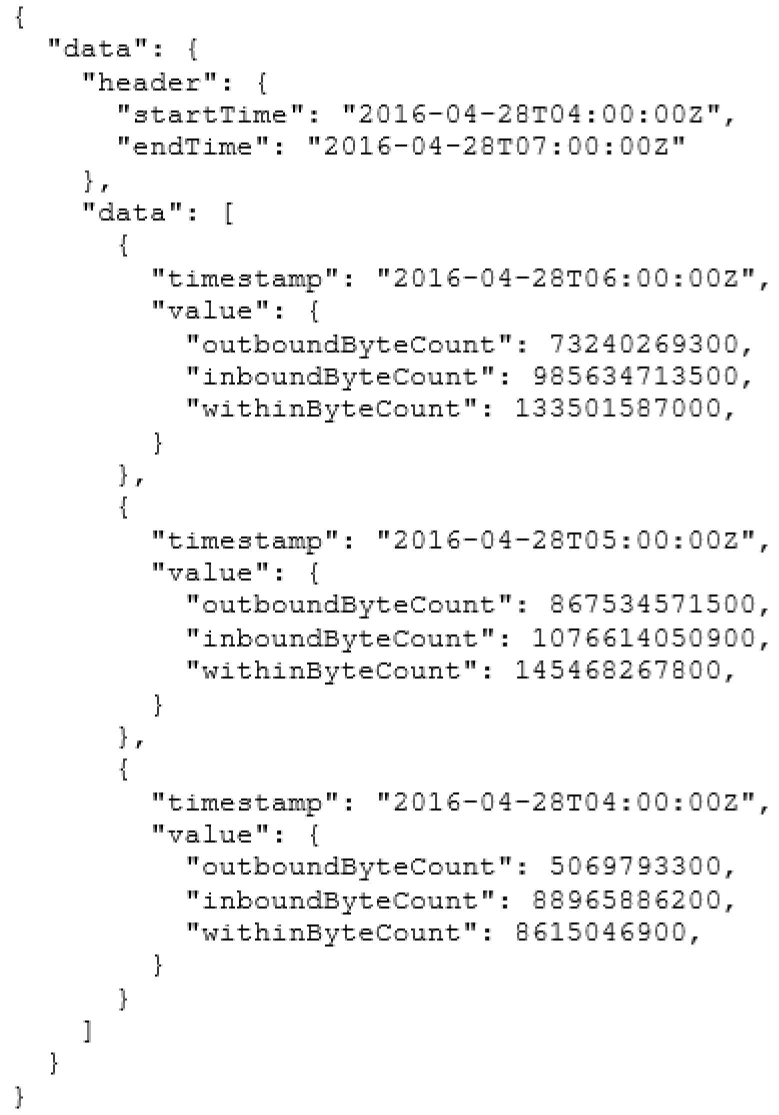
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000,
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800,
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the hourly traffic trend of all applications for a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/applications/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/applications/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/applications/traffic/hourly

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours). The response is an array of traffic trends with each element representing a trend per application.

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.
- applicationId: ID of the application.

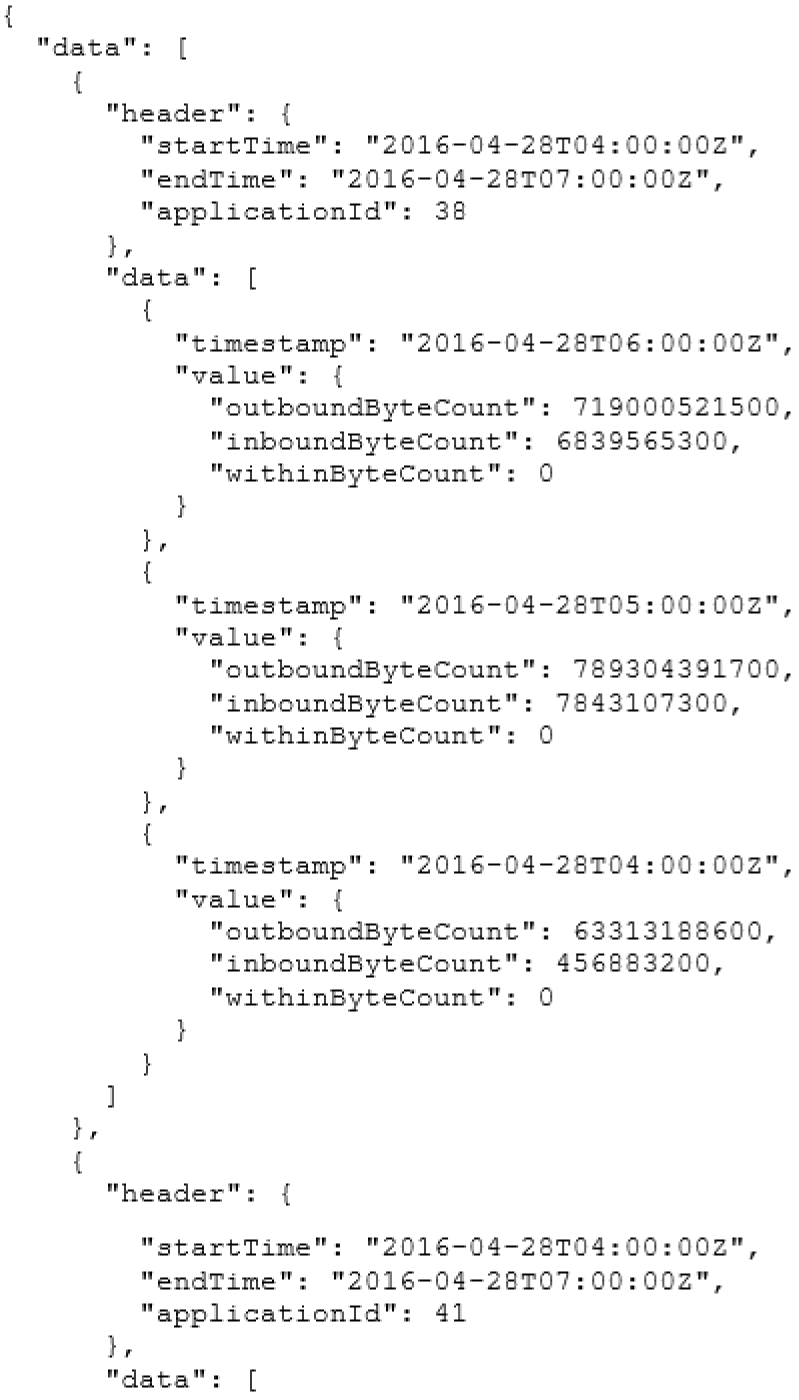
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.

Body

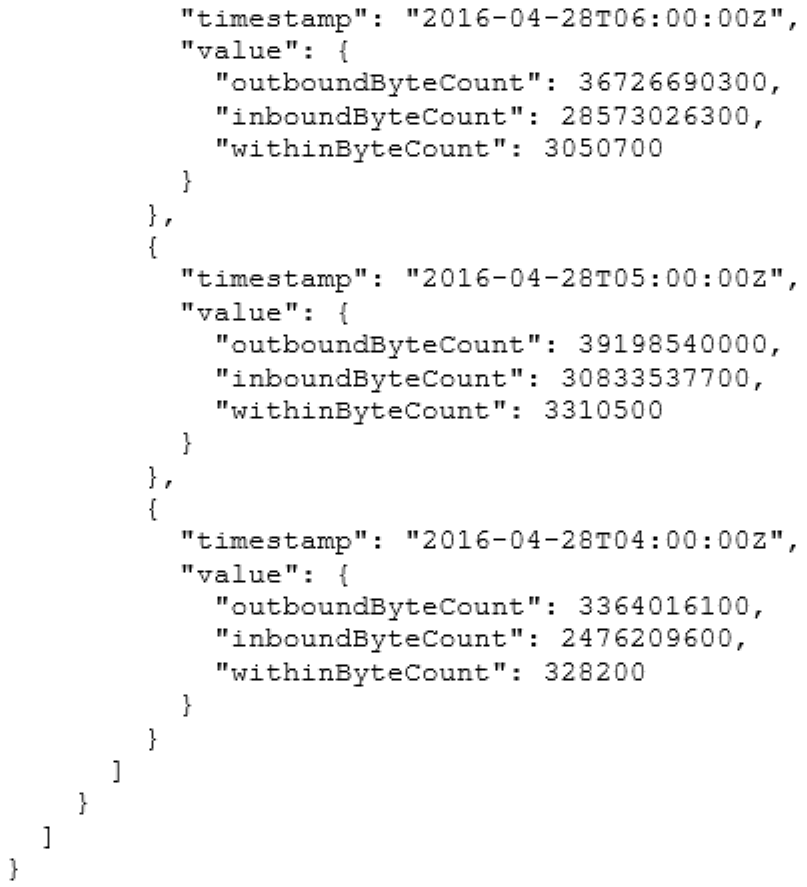
- Type: application/json

Example



```
{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {

        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
      "data": [
```



```
            "timestamp": "2016-04-28T06:00:00Z",
            "value": {
              "outboundByteCount": 36726690300,
              "inboundByteCount": 28573026300,
              "withinByteCount": 3050700
            }
          },
          {
            "timestamp": "2016-04-28T05:00:00Z",
            "value": {
              "outboundByteCount": 39198540000,
              "inboundByteCount": 30833537700,
              "withinByteCount": 3310500
            }
          },
          {
            "timestamp": "2016-04-28T04:00:00Z",
            "value": {
              "outboundByteCount": 3364016100,
              "inboundByteCount": 2476209600,
              "withinByteCount": 328200
            }
          }
        ]
      }
    ]
  }
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters

These APIs retrieve the hourly traffic trend of an application for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/applications/{applicationId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/{applicationId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/{applicationId}/traffic/hourly

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/{applicationId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/applications/{applicationId}/traffic/hourly

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*
- applicationId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

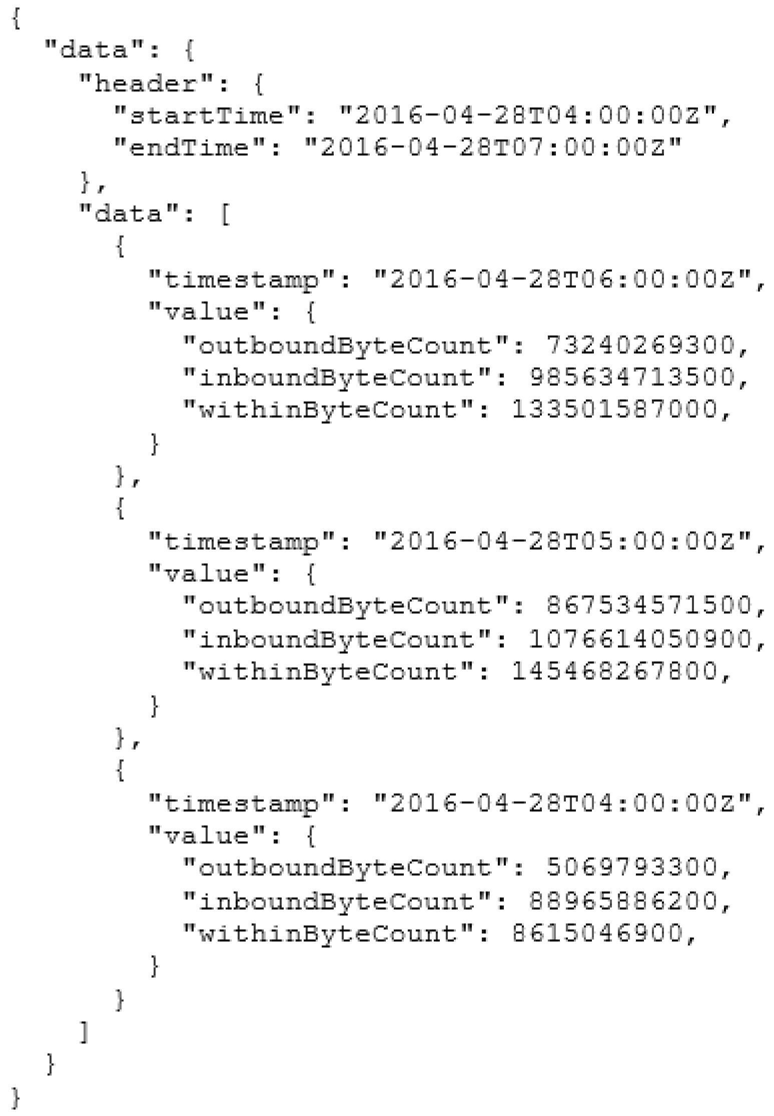
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000,
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800,
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the hourly traffic trend of all application for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/applications/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/applications/traffic/hourly

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours). The response is an array of traffic trends with each element representing a trend per application.

A trend has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.
- applicationId: ID of the application.

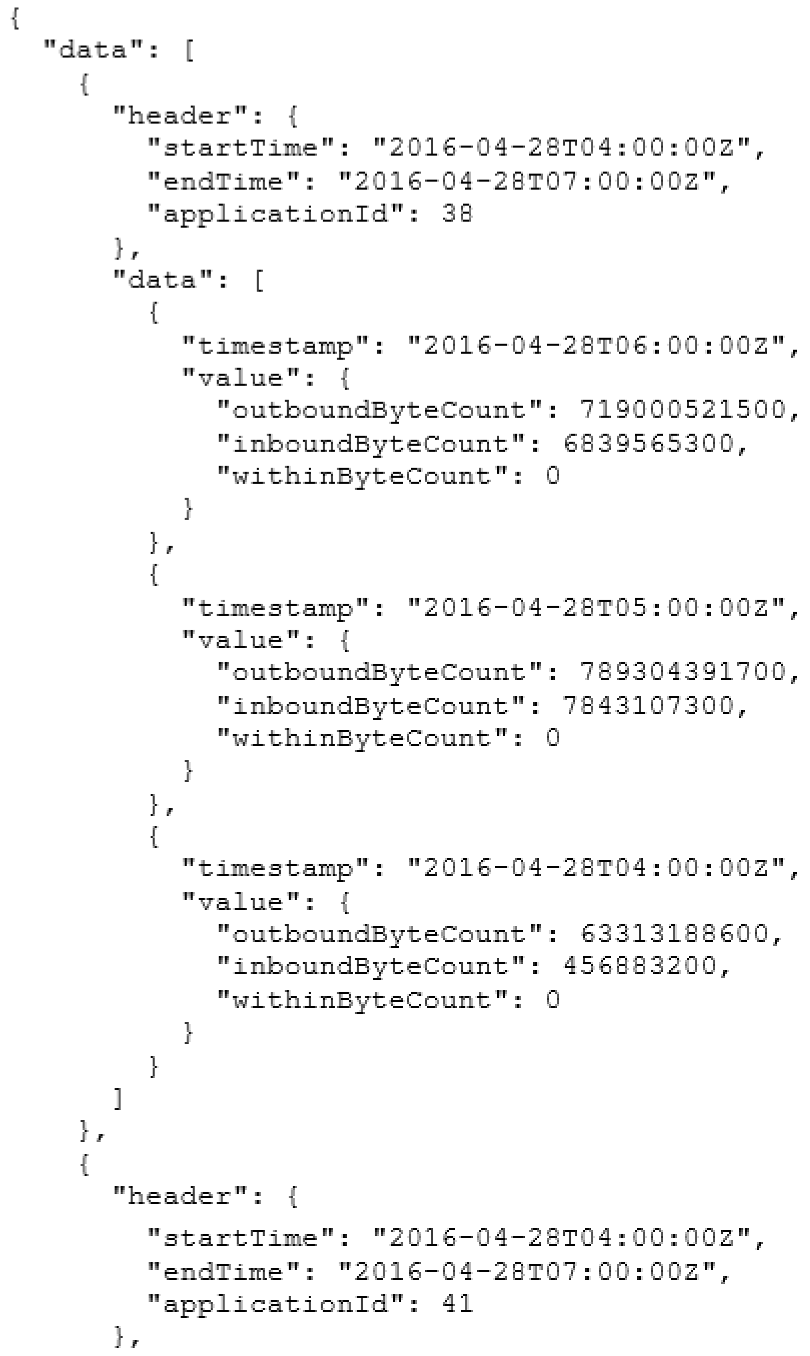
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.

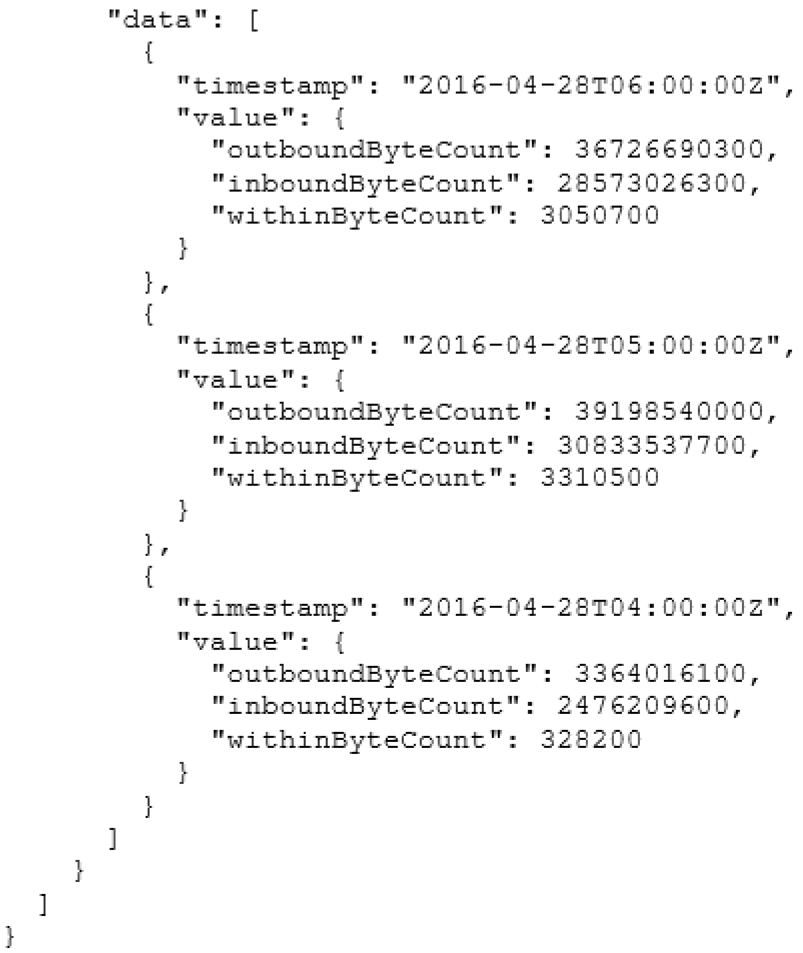
Body

- Type: application/json

Example



```
{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0
          }
        }
      ]
    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T07:00:00Z",
        "applicationId": 41
      },
```



```
      "data": [
        {
          "timestamp": "2016-04-28T06:00:00Z",
          "value": {
            "outboundByteCount": 36726690300,
            "inboundByteCount": 28573026300,
            "withinByteCount": 3050700
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 39198540000,
            "inboundByteCount": 30833537700,
            "withinByteCount": 3310500
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200
          }
        }
      ]
    }
  ]
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the hourly traffic trend for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/traffic/hourly

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/traffic/hourly

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

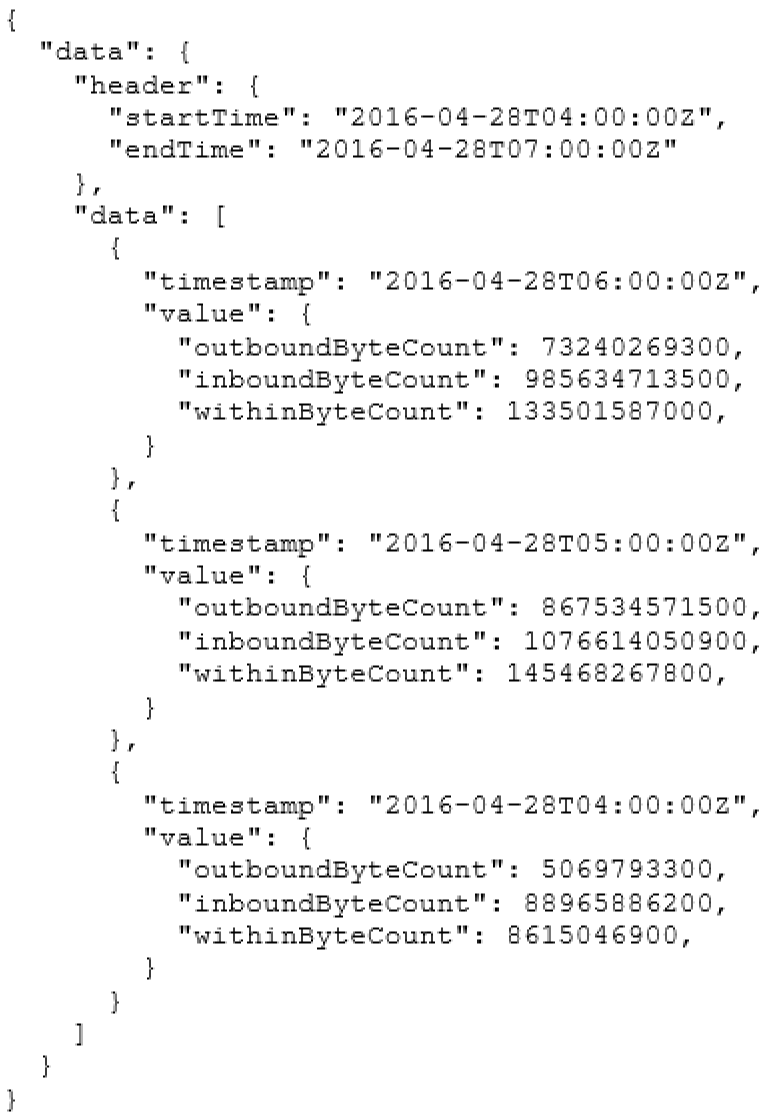
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000,
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800,
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the hourly traffic trend for all Tags in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalgeos/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalhosts/traffic/hourly
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalhosts/traffic/hourly

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is hourly traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

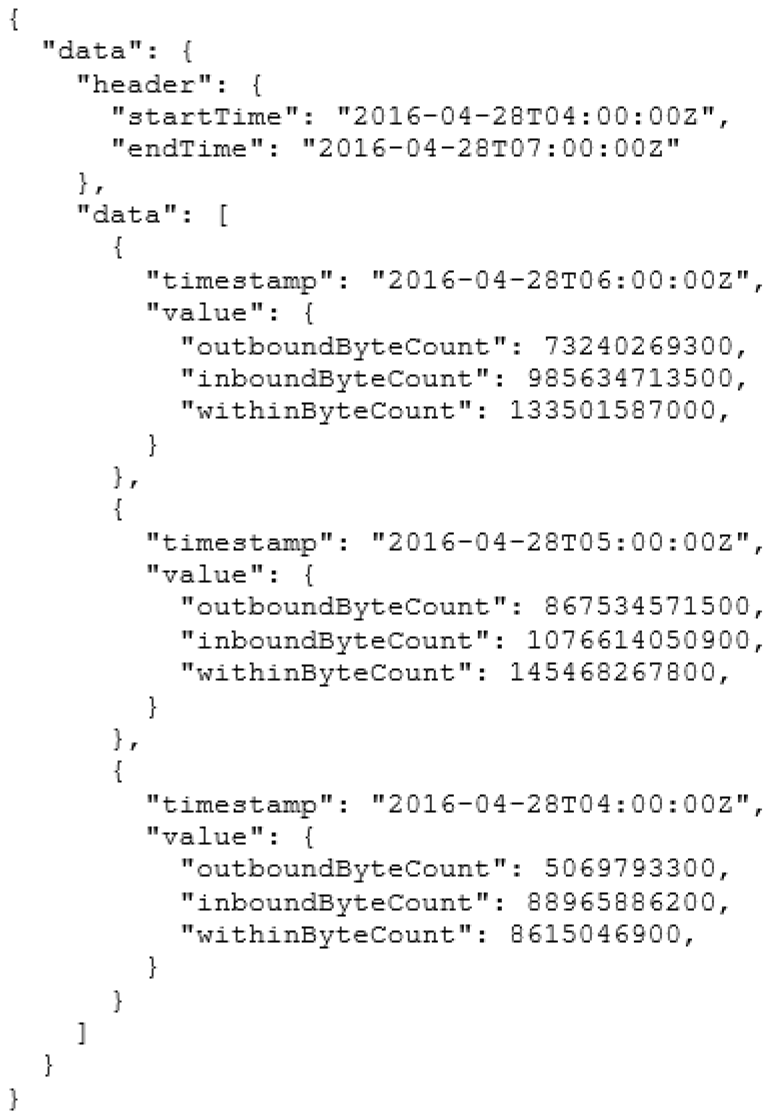
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T07:00:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T06:00:00Z",
        "value": {
          "outboundByteCount": 73240269300,
          "inboundByteCount": 985634713500,
          "withinByteCount": 133501587000,
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 867534571500,
          "inboundByteCount": 1076614050900,
          "withinByteCount": 145468267800,
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters

These APIs retrieve the raw traffic trend of an application for a Tenant:

- hhttps://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/applications/{applicationId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/applications/{applicationId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/applications/{applicationId}/traffic/raw

Request

URI Parameters

- tenantId: *required (integer)*
- applicationId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

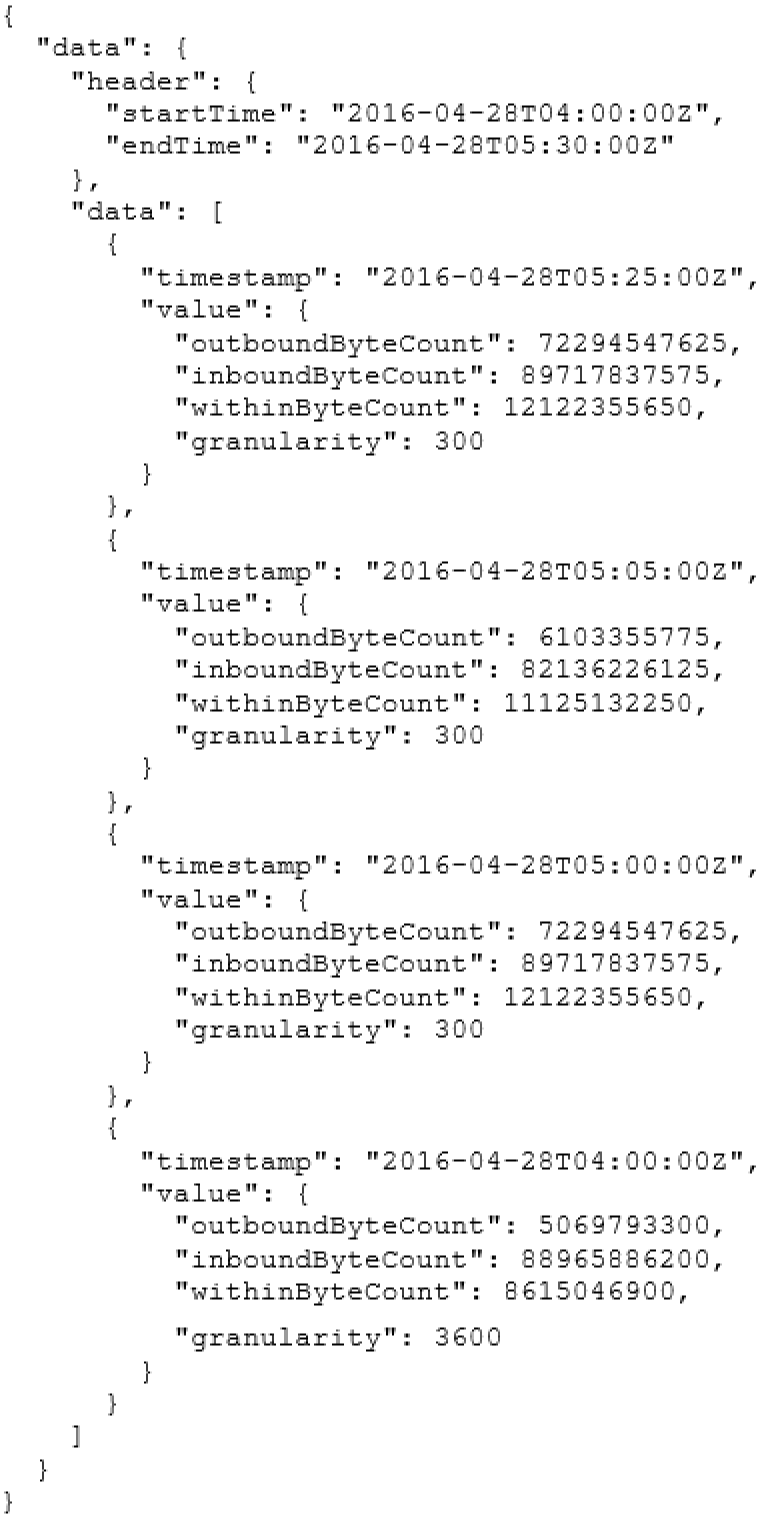
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - granularity: This value is provided only for raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour or 1 day intervals, so this field could have the value of 300, 3600 or 86400.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,

          "granularity": 3600
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the raw traffic trend of all applications for a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/applications/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/applications/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/applications/traffic/raw

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.

- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours). The response is an array of traffic trends with each element representing a trend per application.

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.
- applicationId: ID of the application.

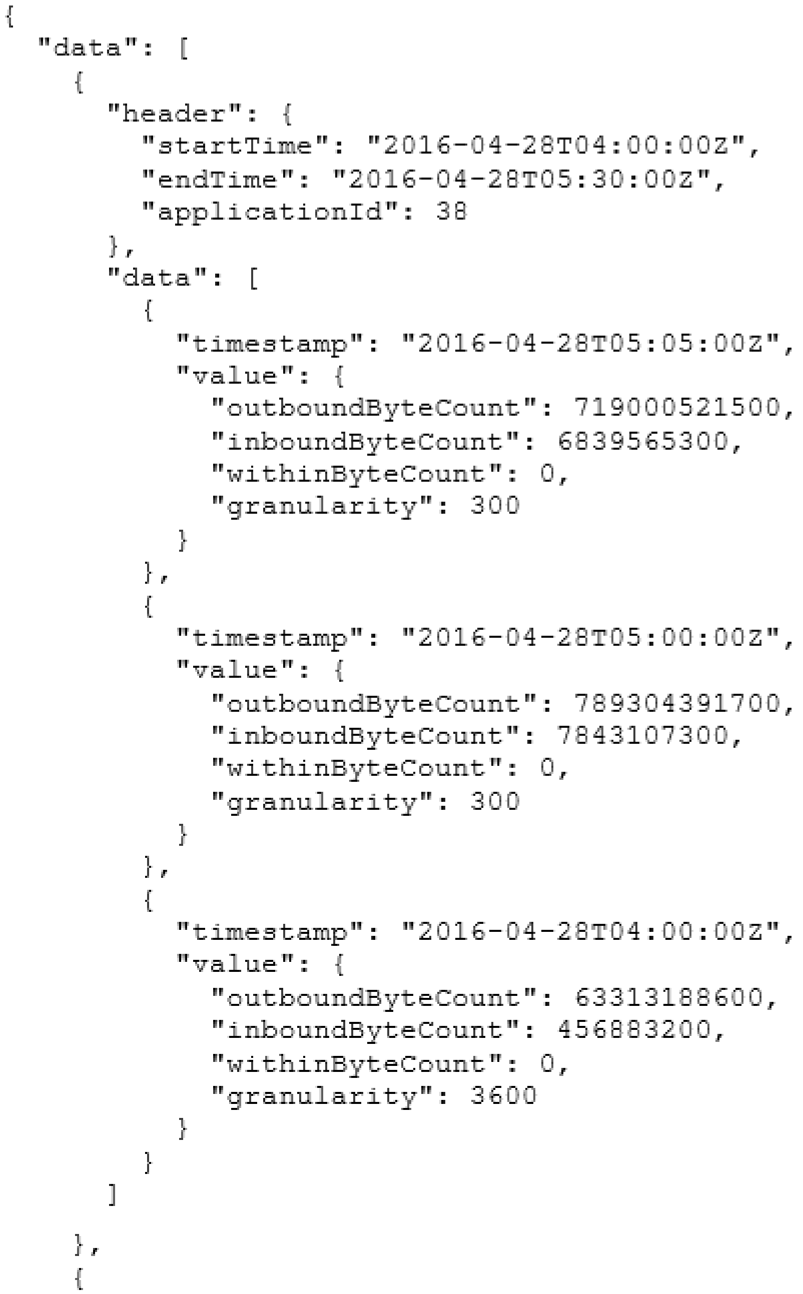
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - granularity: This value is provided only for raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour or 1 day intervals, so this field could have the value of 300, 3600 or 86400.

Body

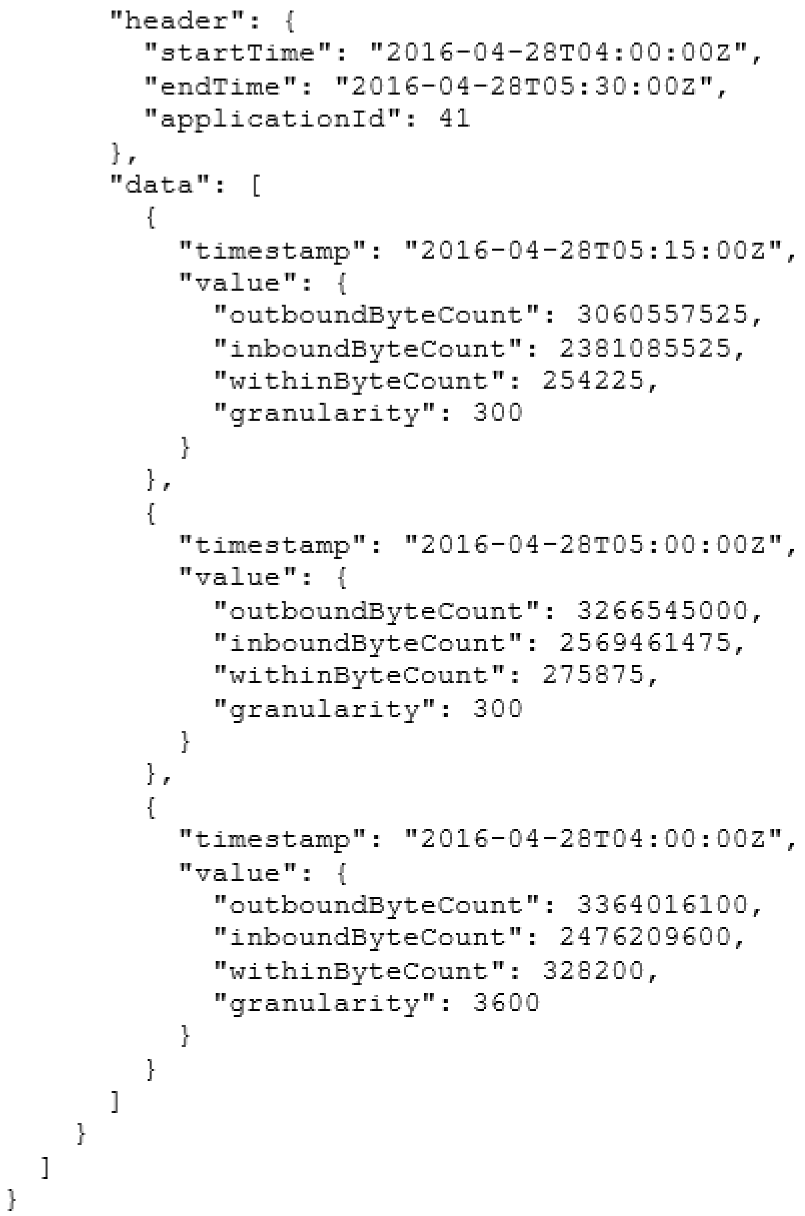
- Type: application/json

Example



```
{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]

    },
    {
```



```
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters

These APIs retrieve the raw traffic trend of an application for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/applications/{applicationId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/{applicationId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/{applicationId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/{applicationId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/applications/{applicationId}/traffic/raw

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*
- application Id: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours). .

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

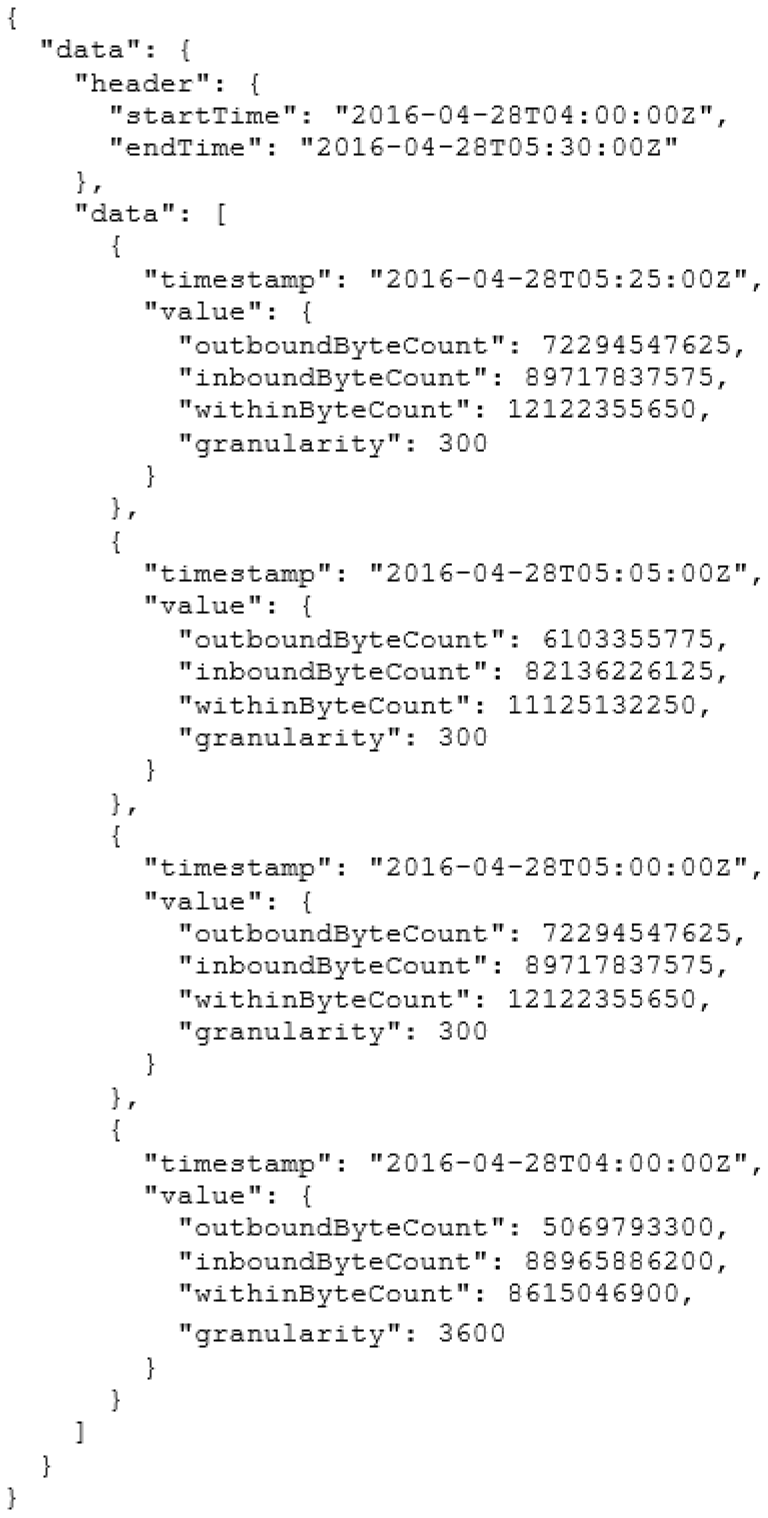
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - granularity: This value is provided only for raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour or 1 day intervals, so this field could have the value of 300, 3600 or 86400.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,
          "granularity": 3600
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the raw traffic trend of all applications for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/applications/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/applications/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/applications/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/applications/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/applications/traffic/raw

  tenantId: *required (integer)*

  tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.

- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours). The response is an array of traffic trends with each element representing a trend per application.

A trend has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.
- applicationId: ID of the application.

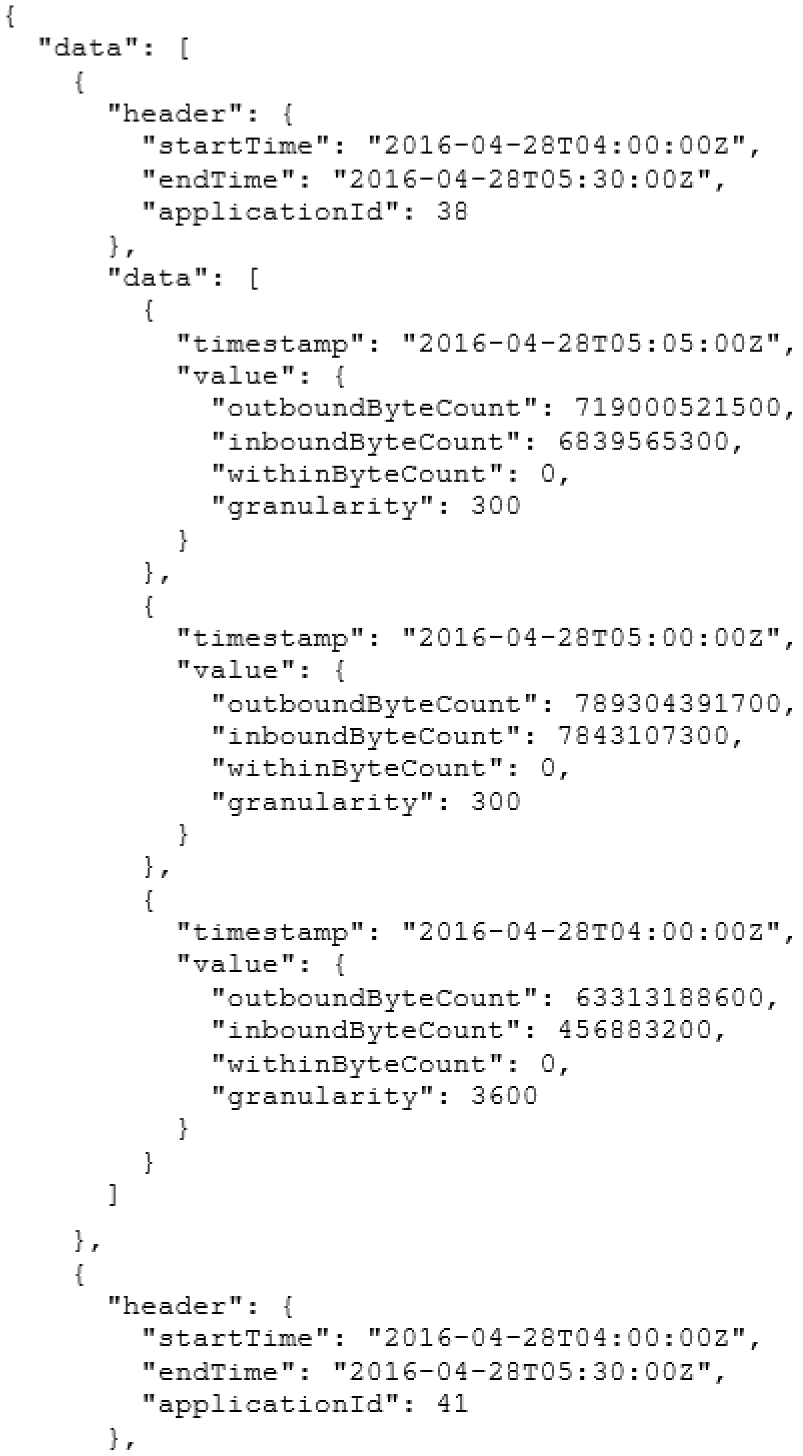
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - granularity: This value is provided only for raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour or 1 day intervals, so this field could have the value of 300, 3600 or 86400.

Body

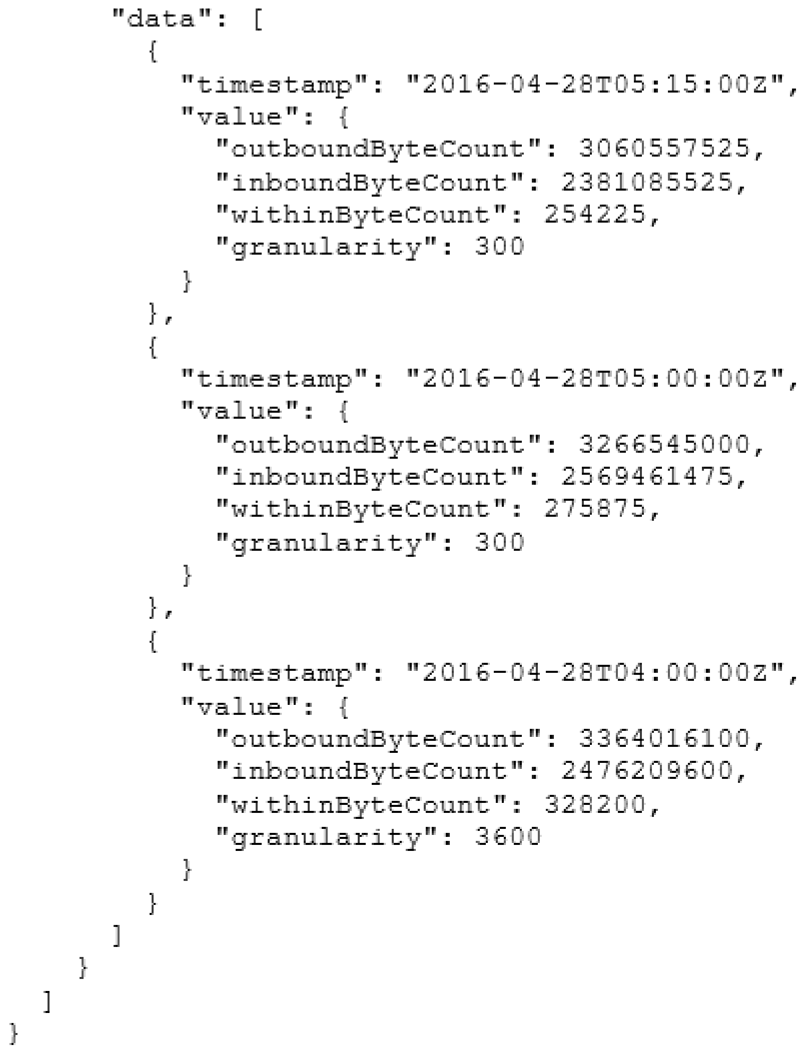
- Type: application/json

Example



```
{
  "data": [
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 38
      },
      "data": [
        {
          "timestamp": "2016-04-28T05:05:00Z",
          "value": {
            "outboundByteCount": 719000521500,
            "inboundByteCount": 6839565300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 789304391700,
            "inboundByteCount": 7843107300,
            "withinByteCount": 0,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 63313188600,
            "inboundByteCount": 456883200,
            "withinByteCount": 0,
            "granularity": 3600
          }
        }
      ]

    },
    {
      "header": {
        "startTime": "2016-04-28T04:00:00Z",
        "endTime": "2016-04-28T05:30:00Z",
        "applicationId": 41
      },
```



```
      "data": [
        {
          "timestamp": "2016-04-28T05:15:00Z",
          "value": {
            "outboundByteCount": 3060557525,
            "inboundByteCount": 2381085525,
            "withinByteCount": 254225,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T05:00:00Z",
          "value": {
            "outboundByteCount": 3266545000,
            "inboundByteCount": 2569461475,
            "withinByteCount": 275875,
            "granularity": 300
          }
        },
        {
          "timestamp": "2016-04-28T04:00:00Z",
          "value": {
            "outboundByteCount": 3364016100,
            "inboundByteCount": 2476209600,
            "withinByteCount": 328200,
            "granularity": 3600
          }
        }
      ]
    }
  ]
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the raw traffic trend for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/traffic/raw

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/traffic/raw

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

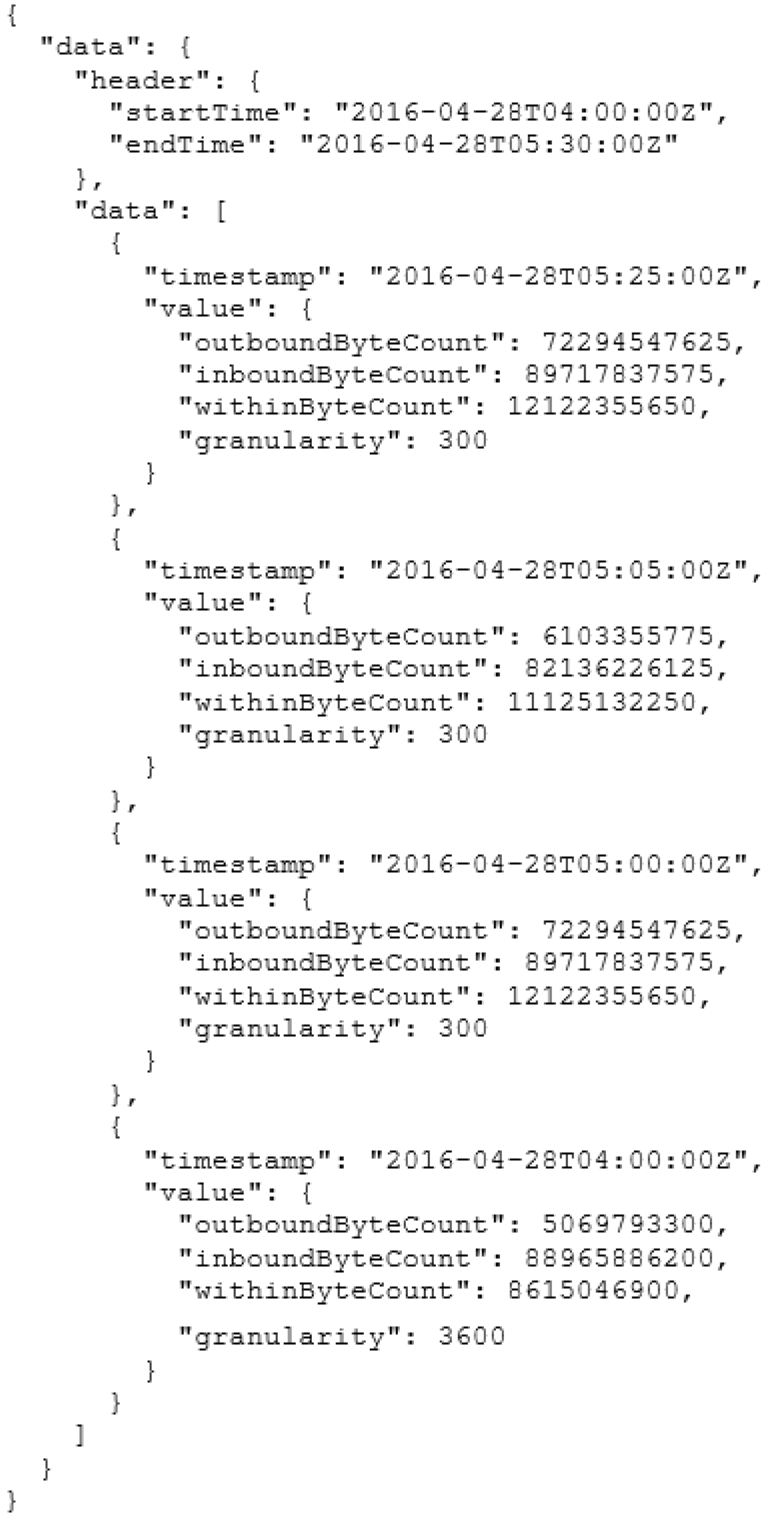
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value.
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - granularity: This value is provided only for raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour or 1 day intervals, so this field could have the value of 300, 3600 or 86400.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,

          "granularity": 3600
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the raw traffic trend for all Tags in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalgeos/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalhosts/traffic/raw
- https://[serviceAddress/sw-reporting/v1/tenants/tenantId/internalhosts/traffic/raw

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is raw traffic trend for 25 hours (current hour and past 24 hours).

The response has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

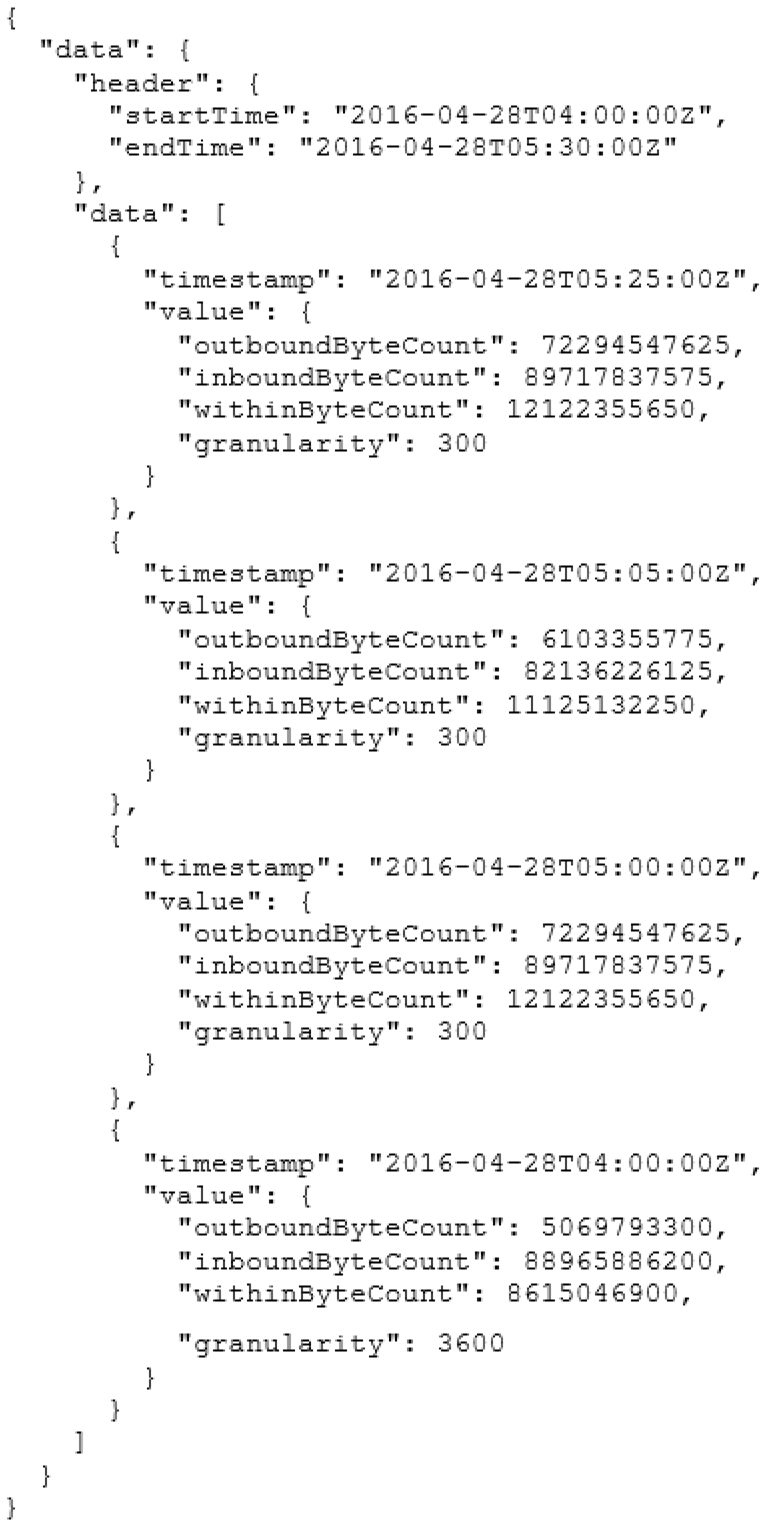
It has a data element which represents the time series data. Each element in the series has the following fields:

- timestamp: For hourly trend, this is the beginning of the hour. For raw trend, this is the beginning of the interval (5 minute, 1 hour or 1 day) which is specified in the value
- value: For traffic trends, this element has the following aggregated values.
  - outboundByteCount: Number of bytes sent by the hosts classified by the tag to hosts not classified by the tag.
  - inboundByteCount: Number of bytes received by the hosts classified by the tag from hosts not classified by the tag.
  - withinByteCount: Number of bytes transmitted (sent/received) within the hosts classified by the tag.
  - granularity: This value is provided only for raw traffic trend and represents the granularity of the trend value in seconds. The raw trend values could be at 5 minute, 1 hour or 1 day intervals, so this field could have the value of 300, 3600 or 86400.

Body

- Type: application/json

Example



```
{
  "data": {
    "header": {
      "startTime": "2016-04-28T04:00:00Z",
      "endTime": "2016-04-28T05:30:00Z"
    },
    "data": [
      {
        "timestamp": "2016-04-28T05:25:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:05:00Z",
        "value": {
          "outboundByteCount": 6103355775,
          "inboundByteCount": 82136226125,
          "withinByteCount": 11125132250,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T05:00:00Z",
        "value": {
          "outboundByteCount": 72294547625,
          "inboundByteCount": 89717837575,
          "withinByteCount": 12122355650,
          "granularity": 300
        }
      },
      {
        "timestamp": "2016-04-28T04:00:00Z",
        "value": {
          "outboundByteCount": 5069793300,
          "inboundByteCount": 88965886200,
          "withinByteCount": 8615046900,

          "granularity": 3600
        }
      }
    ]
  }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters

Top Alarm Host APIs

These APIs retrieve the top alarming hosts for a Tag in a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/customHosts/tags/{tagId}/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/tags/{tagId}/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/tags/{tagId}/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/tags/{tagId}/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/tags/{tagId}/alarms/topHosts

Request

URI Parameters

- tenantId: *required (integer)*
- tagId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.

- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is top alarming hosts for today (since reset hour to now).

A trend has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

It has a data element which represents top alarming hosts data. Each element in the series has the following fields:

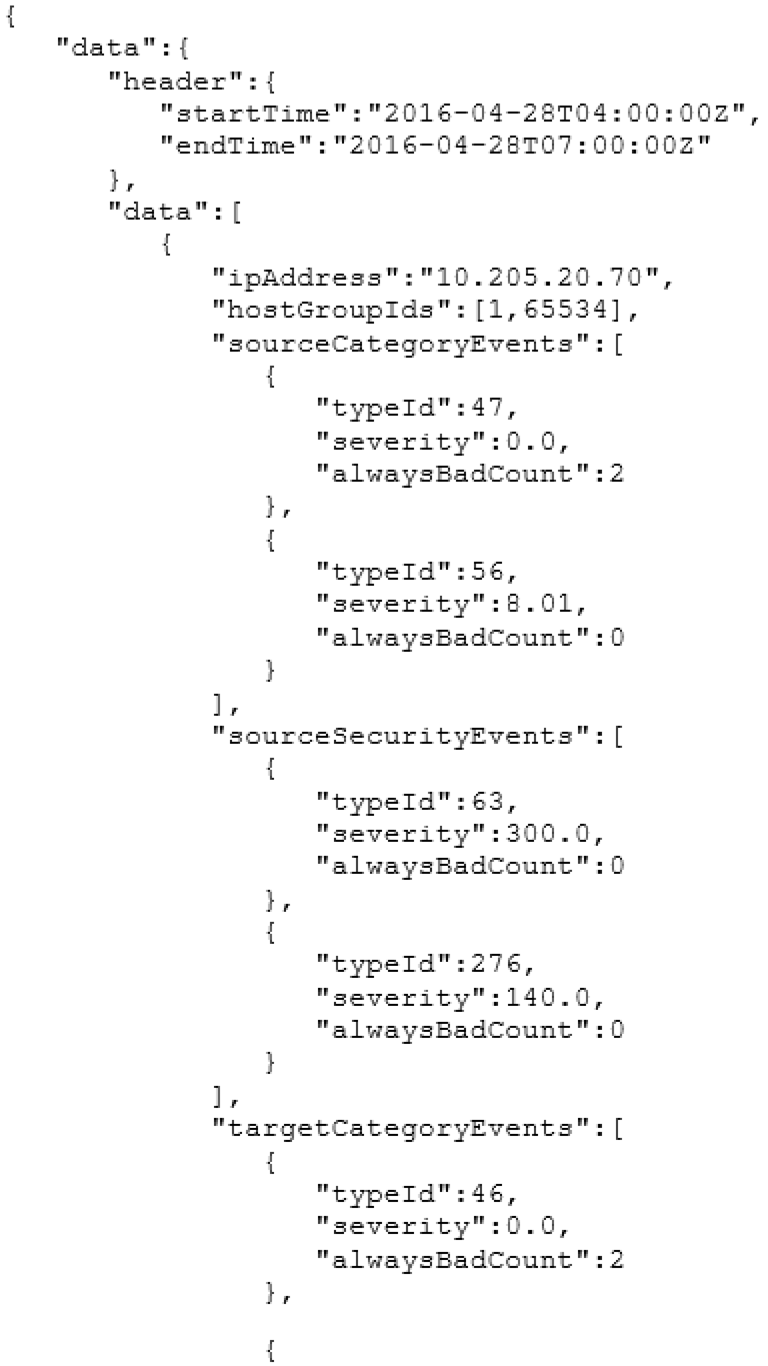
- ipAddress: IP Address of the Host.
- hostGroupIds: Set of host group IDs, which is the union of all the host group IDs associated with the host (source/target) in all the alarms.
- sourceCategoryEvents:
  - typeId: Category event type ID where this host is the source of an alarm.
  - severity: Maximum of the severities for this type of alarm category with the host as the source. Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this alarm category was always bad with the host as the source.
- sourceSecurityEvents:
  - typeId: Security event type ID where this host is the source of a security event.
  - severity: Maximum of the severities for this type of security event with the host as the source. Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this security event was always bad with the host as the source.

- targetCategoryEvents:
  - typeId: Category event type ID where this host is the target of an alarm.
  - severity: Maximum of the severities for this type of alarm category with the host as the target. Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this alarm category was always bad with the host as the target.
- targetSecurityEvents:
  - typeId: Security event type ID where this host is the target of a security event.
  - severity: Maximum of the severities for this type of security event with the host as the target. Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this security event was always bad with the host as the target.

Body

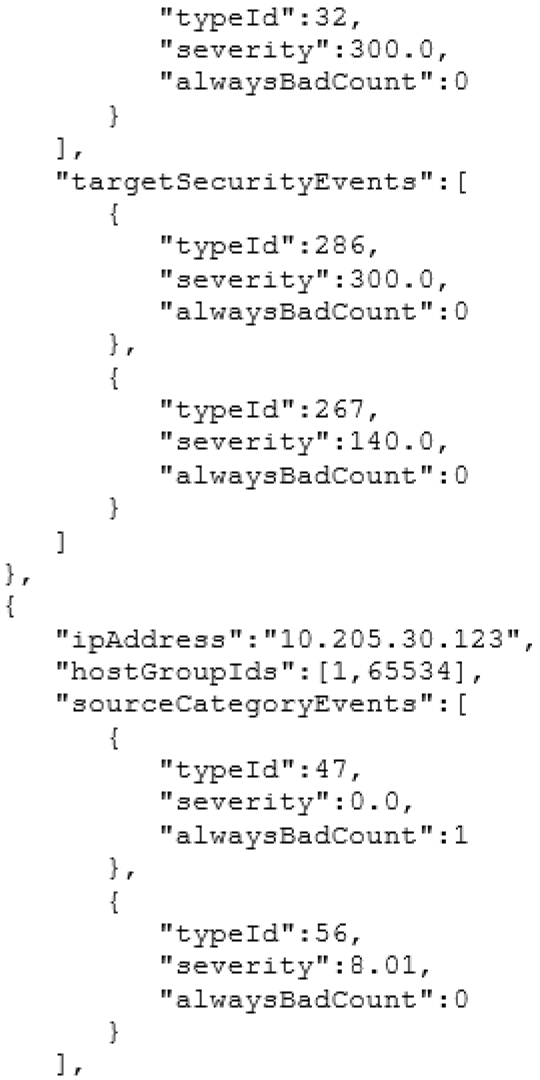
- Type: application/json

Example

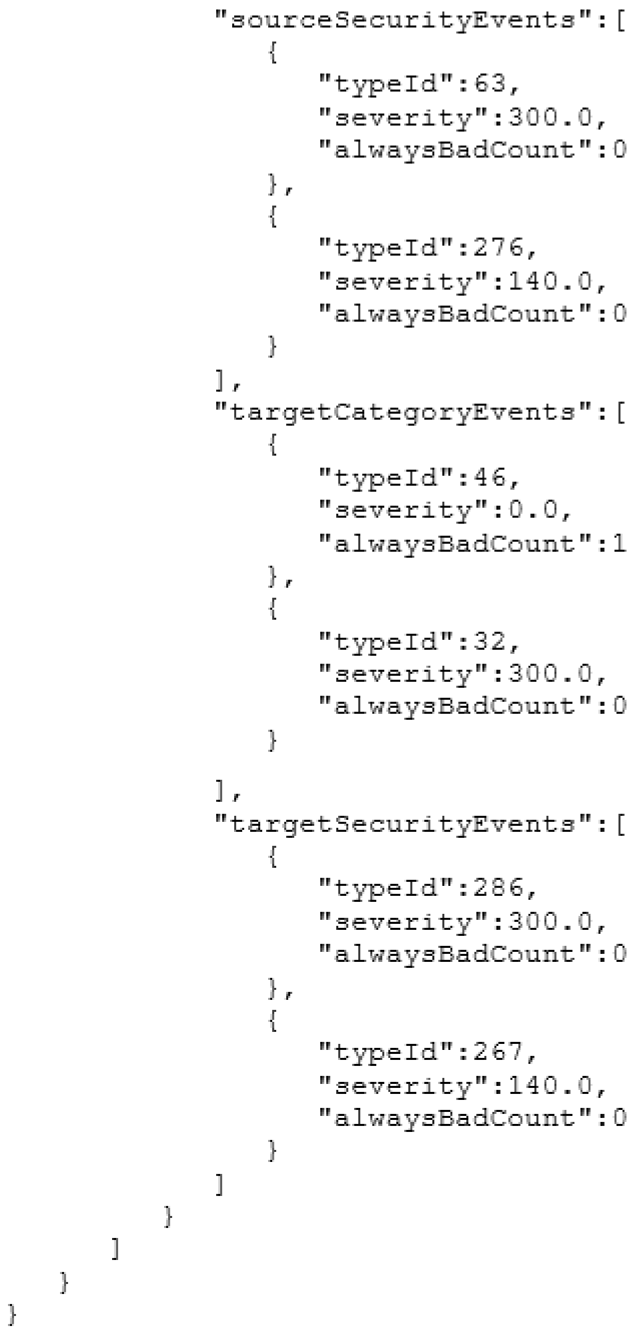


```
{
   "data":{
      "header":{
         "startTime":"2016-04-28T04:00:00Z",
         "endTime":"2016-04-28T07:00:00Z"
      },
      "data":[
         {
            "ipAddress":"10.205.20.70",
            "hostGroupIds":[1,65534],
            "sourceCategoryEvents":[
               {
                  "typeId":47,
                  "severity":0.0,
                  "alwaysBadCount":2
               },
               {
                  "typeId":56,
                  "severity":8.01,
                  "alwaysBadCount":0
               }
            ],
            "sourceSecurityEvents":[
               {
                  "typeId":63,
                  "severity":300.0,
                  "alwaysBadCount":0
               },
               {
                  "typeId":276,
                  "severity":140.0,
                  "alwaysBadCount":0
               }
            ],
            "targetCategoryEvents":[
               {
                  "typeId":46,
                  "severity":0.0,
                  "alwaysBadCount":2
               },

               {
```



```
         "typeId":32,
         "severity":300.0,
         "alwaysBadCount":0
      }
   ],
   "targetSecurityEvents":[
      {
         "typeId":286,
         "severity":300.0,
         "alwaysBadCount":0
      },
      {
         "typeId":267,
         "severity":140.0,
         "alwaysBadCount":0
      }
   ]
},
{
   "ipAddress":"10.205.30.123",
   "hostGroupIds":[1,65534],
   "sourceCategoryEvents":[
      {
         "typeId":47,
         "severity":0.0,
         "alwaysBadCount":1
      },
      {
         "typeId":56,
         "severity":8.01,
         "alwaysBadCount":0
      }
   ],
```



```
            "sourceSecurityEvents":[
               {
                  "typeId":63,
                  "severity":300.0,
                  "alwaysBadCount":0
               },
               {
                  "typeId":276,
                  "severity":140.0,
                  "alwaysBadCount":0
               }
            ],
            "targetCategoryEvents":[
               {
                  "typeId":46,
                  "severity":0.0,
                  "alwaysBadCount":1
               },
               {
                  "typeId":32,
                  "severity":300.0,
                  "alwaysBadCount":0
               }

            ],
            "targetSecurityEvents":[
               {
                  "typeId":286,
                  "severity":300.0,
                  "alwaysBadCount":0
               },
               {
                  "typeId":267,
                  "severity":140.0,
                  "alwaysBadCount":0
               }
            ]
         }
      ]
   }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.

These APIs retrieve the daily alarm trend for a Tenant:

- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalGeos/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalHosts/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/externalThreats/alarms/topHosts
- https://[serviceAddress/sw-reporting/v1/tenants/{tenantId}/internalHosts/alarms/topHosts

Request

URI Parameters

- tenantId: *required (integer)*

Headers

- Cookie: *required (string)*

  JSON Web Token for the authenticated user

Example:
stealthwatch.jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIi
wibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEf
xjoYZgeFONFh7HgQ

Query Parameters

- filter[startAbsolute]: (integer)

  Start time in milliseconds epoch time for the report. Given just this parameter, the end time for the report is the current time.
- filter[endAbsolute]: (integer)

  End time in milliseconds epoch time for the report. This parameter should be provided with either startAbsolute or startRelative.
- filter[startRelative]: (integer)

  Relative start time is in milliseconds. The start time for the report is calculated by subtracting this number from the current time. Given just this parameter, the end time for the report is the current time.
- filter[intervalLength]: (integer)

  Interval is in milliseconds. This parameter is provided with either startAbsolute or startRelative.

Response

HTTP status code 200

If no query parameters are provided, then the response is top alarming hosts for today (since reset hour to now).

A trend has a header element with the following fields:

- startTime: start time for the traffic trend.
- endTime: end time for the traffic trend.

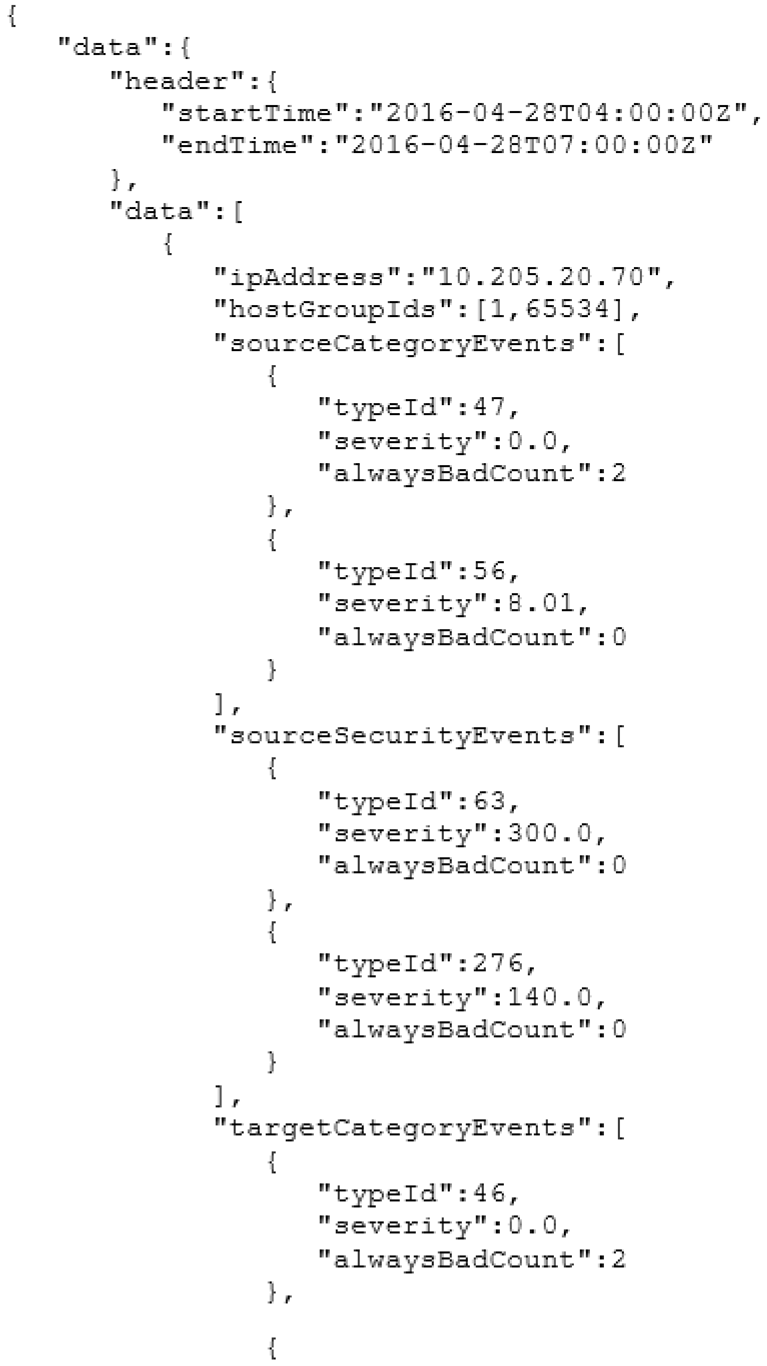
It has a data element which represents top alarming hosts data. Each element in the series has the following fields:

- ipAddress: IP Address of the Host.
- hostGroupIds: Set of host group IDs, which is the union of all the host group IDs associated with the host (source/target) in all the alarms.
- sourceCategoryEvents:
  - typeId: Category event type ID where this host is the source of an alarm.
  - severity: Maximum of the severities for this type of alarm category with the host as the source. Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this alarm category was always bad with the host as the source.
- sourceSecurityEvents:
  - typeId: Security event type ID where this host is the source of a security event.
  - severity: Maximum of the severities for this type of security event with the host as the source. Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this security event was always bad with the host as the source.
- targetCategoryEvents:
  - typeId: Category event type ID where this host is the target of an alarm.
  - severity: Maximum of the severities for this type of alarm category with the host as the target. Severity per alarm is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this alarm category was always bad with the host as the target.
- targetSecurityEvents:
  - typeId: Security event type ID where this host is the target of a security event.
  - severity: Maximum of the severities for this type of security event with the host as the target. Severity per security event is the ratio of the total number of points accumulated divided by the threshold.
  - alwaysBadCount: Number of times the alarm for this security event was always bad with the host as the target.

Body

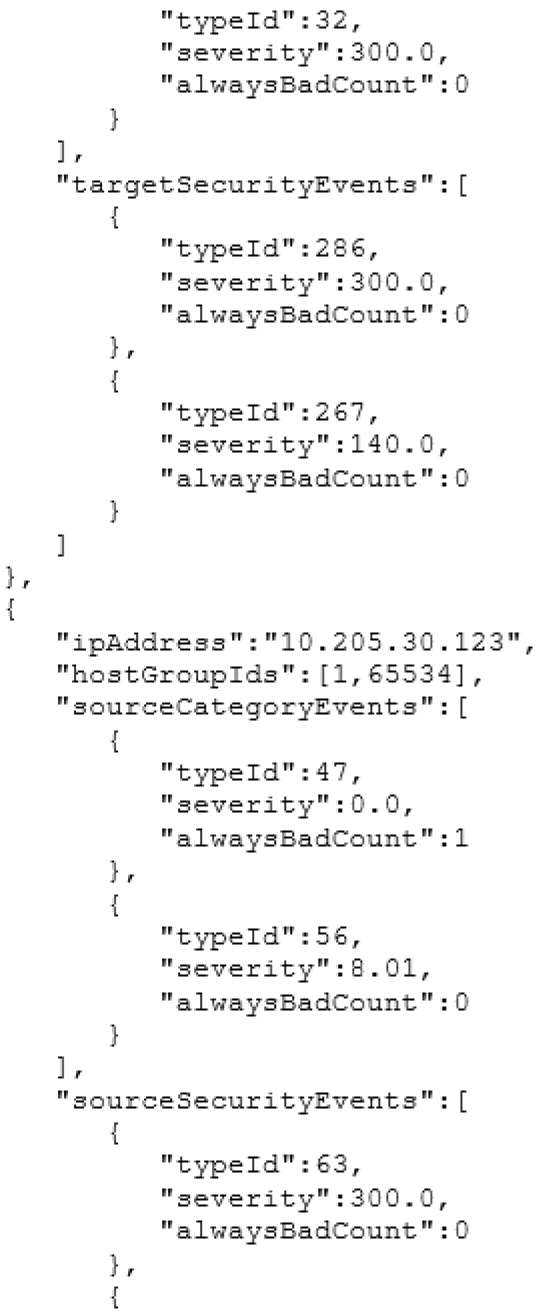
- Type: application/json

Example

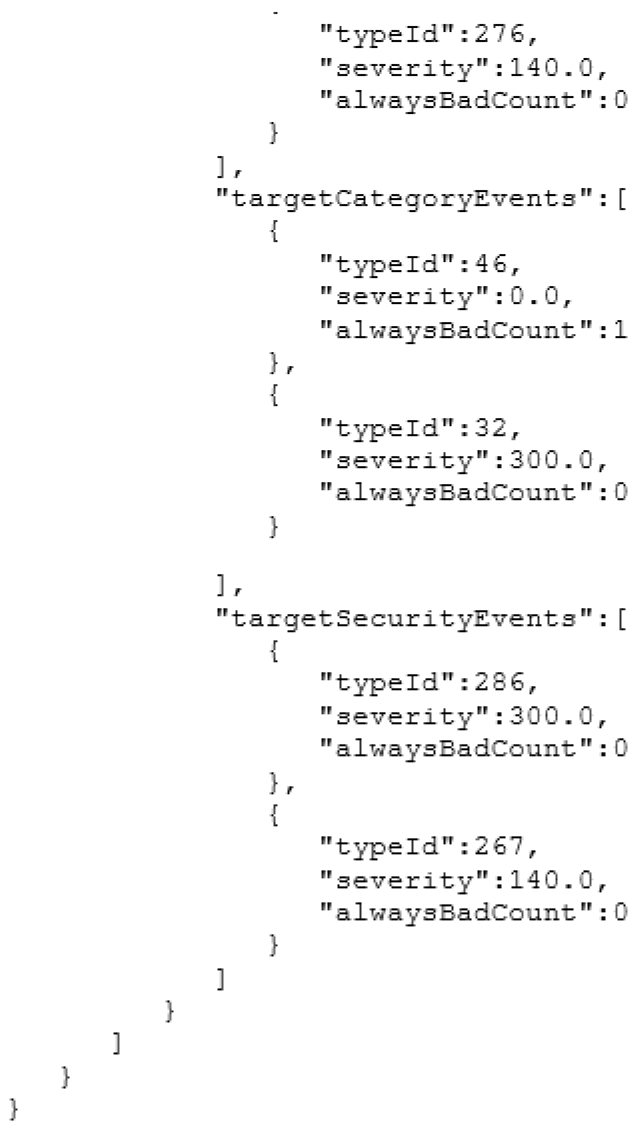


```
{
   "data":{
      "header":{
         "startTime":"2016-04-28T04:00:00Z",
         "endTime":"2016-04-28T07:00:00Z"
      },
      "data":[
         {
            "ipAddress":"10.205.20.70",
            "hostGroupIds":[1,65534],
            "sourceCategoryEvents":[
               {
                  "typeId":47,
                  "severity":0.0,
                  "alwaysBadCount":2
               },
               {
                  "typeId":56,
                  "severity":8.01,
                  "alwaysBadCount":0
               }
            ],
            "sourceSecurityEvents":[
               {
                  "typeId":63,
                  "severity":300.0,
                  "alwaysBadCount":0
               },
               {
                  "typeId":276,
                  "severity":140.0,
                  "alwaysBadCount":0
               }
            ],
            "targetCategoryEvents":[
               {
                  "typeId":46,
                  "severity":0.0,
                  "alwaysBadCount":2
               },

               {
```



```
         "typeId":32,
         "severity":300.0,
         "alwaysBadCount":0
      }
   ],
   "targetSecurityEvents":[
      {
         "typeId":286,
         "severity":300.0,
         "alwaysBadCount":0
      },
      {
         "typeId":267,
         "severity":140.0,
         "alwaysBadCount":0
      }
   ]
},
{
   "ipAddress":"10.205.30.123",
   "hostGroupIds":[1,65534],
   "sourceCategoryEvents":[
      {
         "typeId":47,
         "severity":0.0,
         "alwaysBadCount":1
      },
      {
         "typeId":56,
         "severity":8.01,
         "alwaysBadCount":0
      }
   ],
   "sourceSecurityEvents":[
      {
         "typeId":63,
         "severity":300.0,
         "alwaysBadCount":0
      },
      {
```



```
                  "typeId":276,
                  "severity":140.0,
                  "alwaysBadCount":0
               }
            ],
            "targetCategoryEvents":[
               {
                  "typeId":46,
                  "severity":0.0,
                  "alwaysBadCount":1
               },
               {
                  "typeId":32,
                  "severity":300.0,
                  "alwaysBadCount":0
               }

            ],
            "targetSecurityEvents":[
               {
                  "typeId":286,
                  "severity":300.0,
                  "alwaysBadCount":0
               },
               {
                  "typeId":267,
                  "severity":140.0,
                  "alwaysBadCount":0
               }
            ]
         }
      ]
   }
}
```

HTTP status code 400

Either missing token and client should authenticate, or, invalid parameters.

HTTP status code 401

Expired or invalid token. Client should re-authenticate.

HTTP status code 404

Not Found. Invalid or inaccessible path parameters.