Cisco FXOS CLI 컨피그레이션 가이드, 2.2(2)

초판: 2017년 08월 29일

최종 변경: 2017년 09월 20일

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

목 차

1 장

Firepower Security Appliance 소개

Firepower Security Appliance 정보

Cisco Firepower 4100/9300 섀시는 네트워크 및 콘텐츠 보안 솔루션을 위한 차세대 플랫폼입니다. Firepower 4100/9300 섀시는 Cisco ACI(Application Centric Infrastructure) 보안 솔루션에 포함되며 확장성, 일관된 제어 및 간소화된 관리를 위해 구축된 민첩한 개방형 보안 플랫폼을 제공합니다.

Firepower 4100/9300 섀시에서 제공하는 기능은 다음과 같습니다.

- 모듈형 섀시 기반 보안 시스템 — 고성능의 유연한 입/출력 컨피그레이션 및 확장성을 제공합니다.
- Firepower Chassis Manager — 그래픽 사용자 인터페이스에서는 현재 섀시 상태를 간단하게 시각적으로 표시하며 간소화된 섀시 기능 컨피그레이션을 제공합니다.
- FXOS CLI — 기능 구성, 섀시 상태 모니터링 및 고급 트러블슈팅 기능 액세스를 위해 명령 기반 인터페이스를 제공합니다.
- FXOS REST API — 사용자가 섀시를 프로그래밍 방식으로 구성 및 관리할 수 있습니다.

섀시 상태 모니터링

show environment summary 명령을 사용하여 Firepower 4100/9300 섀시의 전반적인 상태를 보여주는 다음 정보를 확인할 수 있습니다.

- Total power consumption(총 전력 소비량) - 소비된 총 전력량(와트)
- Inlet Temperature(입구 온도) — 시스템 주위 온도(섭씨)
- CPU Temperature(CPU 온도) - 프로세서 온도(섭씨)

- Power supply type(전원 공급장치 유형) - AC 또는 DC
- Power Supply Input Feed Status(전원 공급 장치 입력 공급 상태) — 입력 상태(OK(정상), Fault(결함))
- Power supply output status(전원 공급장치 출력 상태) - 12V 출력 상태(예: OK(정상), Fault(결함))
- Power Supply Overall Status(전원 공급 장치의 전반적인 상태) — PSU의 전반적인 상태(Operable(작동 가능), Removed(제거됨), Thermal problem(열 문제))
- Fan Speed RPM(팬 속도 RPM) — 단일 팬 트레이에 있는 두 팬의 가장 높은 RPM
- Fan speed status(팬 속도 상태) - 팬 속도(Slow(느림), OK(정상), High(빠름), Critical(임계))
- Overall Fan status(전체 팬 상태) - FAN의 전체 상태(Operable(작동 가능), Removed(제거됨), Thermal problem(열 문제))
- Blade Total power consumption(블레이드 총 전력 소비량) - 블레이드의 총 전력 소비량(와트)
- Blade Processor Temperature(블레이드 프로세서 온도) — 보안 모듈/엔진에서 프로세서의 최고 온도(섭씨)

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 섀시 모드를 시작합니다.
Firepower-chassis# **scopechassis1**

단계 **3** 다음 명령을 입력하여 섀시 상태 요약을 확인합니다.
Firepower-chassis /chassis # **show environment summary**

```
Firepower-chassis# scope chassis 1
Firepower-chassis /chassis # show environment summary

Chassis INFO :

Total Power Consumption: 638.000000
Inlet Temperature (C): 32.000000
CPU Temperature (C): 47.000000
Last updated Time: 2017-01-05T23:34:39.115

PSU 1:
Type: AC
Input Feed Status: Ok
12v Output Status: Ok
Overall Status: Operable
PSU 2:
Type: AC
Input Feed Status: Ok
12v Output Status: Ok
Overall Status: Operable

FAN 1
Fan Speed RPM (RPM): 3168
Speed Status: Ok
Overall Status: Operable
FAN 2
```

```
Fan Speed RPM (RPM): 3388
Speed Status: Ok
Overall Status: Operable
FAN 3
Fan Speed RPM (RPM): 3168
Speed Status: Ok
Overall Status: Operable
FAN 4
Fan Speed RPM (RPM): 3212
Speed Status: Ok
Overall Status: Operable

BLADE 1:
Total Power Consumption: 216.000000
Processor Temperature (C): 58.000000
BLADE 2:
Total Power Consumption: 222.000000
Processor Temperature (C): 62.500000
```

2 장

CLI의 개요

관리 객체

FXOS(Firepower eXtensible 운영 체제)에서는 관리 객체 모델을 사용합니다. 여기서 관리 객체란 관리 가능한 물리적 또는 논리적 엔터티를 추상적으로 표현한 것입니다. 예를 들어, 섀시, 보안 모듈, 네트워크 모듈, 포트 및 프로세서는 관리 객체로 표시된 물리적 엔터티이며 라이선스, 사용자 역할 및 플랫폼 정책은 관리 객체로 표시된 논리적 엔터티입니다.

관리 객체에는 연결된 속성(구성 가능)이 하나 이상 있을 수 있습니다.

명령 모드

CLI에는 명령 모드가 계층 구조로 구성되어 있으며, EXEC 모드는 계층 구조에서 최상위 레벨의 모드입니다. 상위 레벨의 모드는 하위 레벨의 모드로 나눠집니다. **create**, **enter**, **scope** 명령을 사용하여 상위 레벨의 모드에서 다음으로 낮은 레벨의 모드로 이동하고 **exit** 명령을 사용하여 모드 계층 구조의 한 레벨 위로 이동합니다. 또한, **top** 명령을 사용하여 모드 계층 구조에서 최상위 레벨로 이동할 수 있습니다.

참고 대부분의 명령 모드는 관리 객체와 연결되어 있으므로 해당 객체와 연결된 모드에 액세스하기 전에 객체를 생성해야 합니다. **create** 및 **enter** 명령을 사용하여 액세스 중인 모드의 관리 객체를 생성합니다. **scope** 명령을 사용하면 관리 객체가 생성되지 않으며 관리 객체가 이미 존재하는 모드에만 액세스할 수 있습니다.

각 모드에는 해당 모드에 입력할 수 있는 명령 집합이 포함됩니다. 각 모드에서 사용할 수 있는 대부분의 명령은 연결된 관리 객체와 관련이 있습니다.

각 모드에 대한 CLI 프롬프트는 현재 모드에 대한 모드 계층 구조의 전체 경로를 보여줍니다. 이는 명령 모드 계층 구조에서 위치를 확인하는 데 도움이 되며 계층 구조를 탐색해야 할 때 매우 유용한 툴이 될 수 있습니다.

다음 표에는 기본 명령 모드, 각 모드에 액세스하는 데 사용되는 명령, 각 모드와 연관된 CLI 프롬프트가 나와 있습니다.

표 1: 기본 명령 모드 및 프롬프트

모드 이름	액세스하는 데 사용되는 명령	모드 프롬프트
EXEC	모든 모드의 **top** 명령	`#`
어댑터	EXEC 모드의 **scope adapter** 명령	`/adapter #`
케이블링	EXEC 모드의 **scope cabling** 명령	`/cabling #`
섀시	EXEC 모드의 **scope chassis** 명령	`/chassis #`
이더넷 서버 도메인	EXEC 모드의 **scope eth-server** 명령. 이 명령 및 모든 하위 명령은 현재 지원되지 않습니다.	`/eth-server #`
이더넷 업링크	EXEC 모드의 **scope eth-uplink** 명령	`/eth-uplink #`
패브릭 인터커넥트	EXEC 모드의 **scope fabric-interconnect** 명령	`/fabric-interconnect #`
펌웨어	EXEC 모드의 **scope firmware** 명령	`/firmware #`
호스트 이더넷 인터페이스	EXEC 모드의 **scope host-eth-if** 명령. 이 명령 및 모든 하위 명령은 이 레벨에서 지원되지 않습니다. 호스트 이더넷 인터페이스 명령은 `/adapter #` 모드에서 사용할 수 있습니다.	`/host-eth-if #`
라이선스	EXEC 모드의 **scope license** 명령	`/license #`

모드 이름	액세스하는 데 사용되는 명령	모드 프롬프트
모니터링	EXEC 모드의 **scope monitoring** 명령	`/monitoring #`
조직	EXEC 모드의 **scope org** 명령	`/org #`
패킷 캡처	EXEC 모드의 **scope packet-capture** 명령	`/packet-capture #`
보안	EXEC 모드의 **scope security** 명령	`/security #`
서버	EXEC 모드의 **scope server** 명령	`/server #`
서비스 프로파일	EXEC 모드의 **scope service-profile** 명령. 서비스 프로파일을 변경하거나 구성하지 마십시오. 즉. **create**, **set** 또는 **delete** 하위 명령 집합을 사용하지 마십시오.	`/service-profile #`
ssa	EXEC 모드의 **scope ssa** 명령	`/ssa #`
시스템	EXEC 모드의 **scope system** 명령	`/system #`
가상 HBA	EXEC 모드의 **scope vhba** 명령. 이 명령 및 모든 하위 명령은 현재 지원되지 않습니다.	`/vhba #`
가상 NIC	EXEC 모드의 **scope vnic** 명령	`/vnic #`

객체 명령

객체 관리에 사용 가능한 일반 명령 4개가 있습니다.

- **create** *object*
- **delete** *object*
- **enter** *object*
- **scope** *object*

영구 객체 또는 사용자가 인스턴스화한 객체 등 모든 관리 객체에 **scope** 명령을 사용할 수 있습니다. 나머지 명령을 사용하여 사용자가 인스턴스화한 객체를 생성하고 관리할 수 있습니다. 모든 **create** *object* 명령에는 해당하는 **delete** *object* 및 **enter** *object* 명령이 있습니다.

사용자가 인스턴스화한 객체 관리 시 이러한 명령의 행동은 다음 표에 설명된 대로 객체의 유무에 따라 달라집니다.

표 2: 객체가 없는 경우의 일반적인 행동

명령	행동
create *object*	객체가 생성되고 해당하는 경우 컨피그레이션 모드가 시작됩니다.
delete *object*	오류 메시지가 생성됩니다.
enter *object*	객체가 생성되고 해당하는 경우 컨피그레이션 모드가 시작됩니다.
scope *object*	오류 메시지가 생성됩니다.

표 3: 객체가 있는 경우의 일반적인 행동

명령	행동
create *object*	오류 메시지가 생성됩니다.
delete *object*	객체가 삭제됩니다.
enter *object*	해당하는 경우 객체의 컨피그레이션 모드가 시작됩니다.
scope *object*	객체의 컨피그레이션 모드가 시작됩니다.

명령 완성

아무 모드에서나 **Tab** 키를 사용하여 명령을 완성할 수 있습니다. 명령 이름의 일부를 입력하고 **Tab** 키를 누르면 전체 명령이 표시되거나 다른 키워드 또는 인수 값을 입력해야 하는 지점까지 표시됩니다.

명령 기록

CLI는 현재 세션에서 사용되는 모든 명령을 저장합니다. 위쪽 화살표 또는 아래쪽 화살표 키를 사용하여 이전에 사용한 명령을 하나씩 살펴볼 수 있습니다. 위쪽 화살표 키를 누르면 저장된 이전 명령으로 이동하고 아래쪽 화살표 키를 누르면 저장된 다음 명령으로 이동합니다. 저장된 마지막 명령에 도달하면 아래쪽 화살표 키를 눌러도 아무 명령도 실행되지 않습니다.

간단히 저장된 명령을 하나씩 살펴보고 해당 명령을 불러온 다음 **Enter** 키를 눌러 저장된 명령을 다시 입력할 수 있습니다. 명령은 사용자가 수동으로 입력한 것처럼 입력됩니다. **Enter** 키를 누르기 전에 명령을 불러 변경할 수도 있습니다.

보류 중인 명령 커밋, 삭제 및 보기

CLI에서 컨피그레이션 명령을 입력하는 경우 **commit-buffer** 명령을 입력할 때까지 해당 명령이 적용되지 않습니다. 커밋될 때까지 컨피그레이션 명령은 보류되며 **discard-buffer** 명령을 입력하여 삭제할 수 있습니다.

여러 명령 모드에서 보류 중인 변경 사항을 누적한 다음 **commit-buffer** 명령으로 한꺼번에 적용할 수 있습니다. 모든 명령 모드에서 **show configuration pending** 명령을 입력하여 보류 중인 명령을 확인할 수 있습니다.

참고 여러 명령을 함께 커밋하는 것은 원자 조작이 아닙니다. 실패하는 명령이 있는 경우에도 그 외에는 성공적인 명령이 적용되며, 장애가 발생한 명령은 오류 메시지로 보고됩니다.

보류 중인 명령이 있는 경우 별표(*)가 명령 프롬프트 앞에 나타납니다. 이 별표는 **commit-buffer** 명령을 입력하면 사라집니다.

다음 예는 프롬프트가 명령 입력 프로세스 중 어떻게 변경되는지 보여줍니다.

```
Firepower# scope system
Firepower /system # scope services
Firepower /system/services # create ntp-server 192.168.200.101
Firepower /system/services* # show configuration pending
 scope services
+    create ntp-server 192.168.200.101
 exit
Firepower /system/services* # commit-buffer
Firepower /system/services #
```

CLI용 온라인 도움말

언제든지 ? 문자를 입력하면 명령 구문의 현재 상태에서 사용 가능한 옵션이 표시됩니다.

프롬프트에 아무것도 입력하지 않고 ?를 입력하면 현재 모드에서 사용 가능한 명령이 모두 나열됩니다. 명령을 일부 입력하고 ?를 입력하면 명령 구문의 현재 위치에서 사용 가능한 모든 키워드 및 인수가 나열됩니다.

CLI 세션 제한

Firepower eXtensible 운영 체제에서는 한 번에 활성화할 수 있는 CLI 세션의 수가 총 32개로 제한됩니다. 이 값은 구성할 수 없습니다.

3 장

시작하기

작업 플로우

다음 절차에서는 Firepower 4100/9300 섀시 구성 시 완료해야 하는 기본 작업을 보여줍니다.

절차

단계 **1**	Firepower 4100/9300 섀시 하드웨어를 구성합니다(Cisco Firepower Security Appliance 하드웨어 설치 가이드 참조).
단계 **2**	초기 컨피그레이션을 완료합니다(초기 컨피그레이션, 12 페이지 참조).
단계 **3**	날짜 및 시간을 설정합니다(날짜 및 시간 설정, 99 페이지 참조).
단계 **4**	DNS 서버를 구성합니다(DNS 서버 구성, 136 페이지 참조).
단계 **5**	제품 라이선스를 등록합니다(ASA용 라이선스 관리, 17 페이지 참조).
단계 **6**	사용자를 구성합니다(사용자 관리, 31 페이지 참조).
단계 **7**	필요 시 소프트웨어 업데이트를 수행합니다(이미지 관리, 53 페이지 참조).
단계 **8**	추가 플랫폼 설정을 구성합니다(플랫폼 설정, 99 페이지 참조).
단계 **9**	인터페이스를 구성합니다(인터페이스 관리, 139 페이지 참조).
단계 **10**	논리적 디바이스를 생성합니다(논리적 디바이스, 147 페이지 참조).

초기 컨피그레이션

Firepower Chassis Manager 또는 FXOS CLI를 사용하여 시스템을 구성하고 관리할 수 있으려면 먼저 콘솔 포트를 통해 액세스하는 FXOS CLI를 사용하여 초기 컨피그레이션 작업 일부를 수행해야 합니다. FXOS CLI를 사용하여 처음으로 Firepower 4100/9300 섀시에 액세스할 때 시스템을 구성하는 데 사용할 수 있는 설정 마법사가 나타납니다.

기존 백업 파일에서 시스템 컨피그레이션을 복원하거나 설정 마법사를 통해 수동으로 시스템을 설정하도록 선택할 수 있습니다. 시스템 복원을 선택할 경우 관리 네트워크에서 백업 파일에 접근할 수 있어야 합니다.

Firepower 4100/9300 섀시의 단일 관리 포트에 대해 단 하나의 IPv4 주소, 게이트웨이, 서브넷 마스크 또는 단 하나의 IPv6 주소, 게이트웨이, 네트워크 접두사만 지정해야 합니다. 관리 포트 IP 주소로 IPv4 또는 IPv6 주소 중 하나를 구성할 수 있습니다.

시작하기 전에

1. Firepower 4100/9300 섀시에서 다음 물리적 연결을 확인합니다.
   - 콘솔 포트가 컴퓨터 터미널 또는 콘솔 서버에 물리적으로 연결되어 있습니다.
   - 1Gbps 이더넷 관리 포트가 외부 허브, 스위치 또는 라우터에 연결되어 있습니다.

   자세한 내용은 Cisco Firepower Security Appliance 하드웨어 설치 가이드를 참조하십시오.
2. 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 파라미터가 다음과 같은지 확인합니다.
   - 9600 보(baud)
   - 8 데이터 비트
   - 패리티 없음
   - 1 스톱 비트

절차

단계 **1** 콘솔 포트에 연결합니다.

단계 **2** Firepower 4100/9300 섀시의 전원을 켭니다.
Firepower 4100/9300 섀시가 부팅될 때 전원 켜짐 자가 테스트 메시지가 표시됩니다.

단계 **3** 구성되지 않은 시스템을 부팅하는 경우, 설정 마법사에서 시스템을 구성하는 데 필요한 다음 정보를 묻는 프롬프트를 표시합니다.
- 설정 모드(전체 시스템 백업에서 복원 또는 초기 설정)
- 강력한 비밀번호 시행 정책(강력한 비밀번호 지침에 대해서는 사용자 어카운트, 32 페이지 참조)

- 관리자 비밀번호
- 시스템 이름
- 관리 포트 IPv4 주소 및 서브넷 마스크 또는 IPv6 주소 및 접두사
- 기본 게이트웨이 IPv4 또는 IPv6 주소
- DNS 서버 IPv4 또는 IPv6 주소
- 기본 도메인 이름

단계 **4** 설정 요약을 검토하고 **yes**를 입력하여 설정을 저장하고 적용하거나 **no**를 입력하여 다시 설정 마법사를 통해 일부 설정을 변경합니다.
설정 마법사를 다시 사용하도록 선택하는 경우 이전에 입력한 값이 대괄호에 나타납니다. 이전에 입력한 값을 승인하려면 **Enter** 키를 누릅니다.

다음 예에서는 IPv4 관리 주소를 사용하여 컨피그레이션을 설정합니다.

```
Enter the setup mode; setup newly or restore from backup. (setup/restore) ? setup
You have chosen to setup a new Fabric interconnect. Continue? (y/n): y
Enforce strong password? (y/n) [y]: n
Enter the password for "admin": adminpassword%958
Confirm the password for "admin": adminpassword%958
Enter the system name: foo
Physical Switch Mgmt0 IP address : 192.168.10.10
Physical Switch Mgmt0 IPv4 netmask: 255.255.255.0
IPv4 address of the default gateway: 192.168.10.1
Configure the DNS Server IP address? (yes/no) [n]: yes
  DNS IP address: 20.10.20.10
Configure the default domain name? (yes/no) [n]: yes
  Default domain name: domainname.com
Following configurations will be applied:
  Switch Fabric=A
  System Name=foo
  Enforce Strong Password=no
  Physical Switch Mgmt0 IP Address=192.168.10.10
  Physical Switch Mgmt0 IP Netmask=255.255.255.0
  Default Gateway=192.168.10.1
  IPv6 value=0
  DNS Server=20.10.20.10
  Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

다음 예에서는 IPv6 관리 주소를 사용하여 컨피그레이션을 설정합니다.

```
Enter the setup mode; setup newly or restore from backup. (setup/restore) ? setup
You have chosen to setup a new Fabric interconnect. Continue? (y/n): y
Enforce strong password? (y/n) [y]: n
Enter the password for "admin": adminpassword%652
Confirm the password for "admin": adminpassword%652
Enter the system name: foo
Physical Switch Mgmt0 IP address :  2001::107
Physical Switch Mgmt0 IPv6 prefix: 64
IPv6 address of the default gateway: 2001::1
Configure the DNS Server IPv6 address? (yes/no) [n]: yes
  DNS IP address: 2001::101
Configure the default domain name? (yes/no) [n]: yes
  Default domain name: domainname.com
Following configurations will be applied:
  Switch Fabric=A
  System Name=foo
  Enforced Strong Password=no
  Physical Switch Mgmt0 IPv6 Address=2001::107
```

```
  Physical Switch Mgmt0 IPv6 Prefix=64
  Default Gateway=2001::1
  Ipv6 value=1
  DNS Server=2001::101
  Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

FXOS CLI 액세스

콘솔 포트에 연결된 터미널을 사용하여 FXOS CLI에 연결할 수 있습니다. 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 파라미터가 다음과 같은지 확인합니다.

- 9600 보(baud)
- 8 데이터 비트
- 패리티 없음
- 1 스톱 비트

SSH 및 텔넷을 사용하여 FXOS CLI에 연결할 수도 있습니다. Firepower eXtensible 운영 체제는 최대 8개의 동시 SSH 연결을 지원합니다. SSH를 사용하여 연결하려면 Firepower 4100/9300 섀시의 IP 주소 또는 호스트 이름을 알아야 합니다.

다음 구문 예시 중 하나를 사용하여 SSH, 텔넷 또는 Putty로 로그인할 수 있습니다.

참고 SSH 로그인에서는 대/소문자를 구분합니다.

SSH를 사용하는 Linux 터미널에서 다음 구문을 사용합니다.

- **sshucs-***auth-domain**username*@{*UCSM-ip-address*|*UCMS-ipv6-address*}

```
ssh ucs-example\\jsmith@192.0.20.11
ssh ucs-example\\jsmith@2001::1
```

- **ssh -lucs-***auth-domain**username* {*UCSM-ip-address*| *UCSM-ipv6-address*| *UCSM-host-name*}

```
ssh -l ucs-example\\jsmith 192.0.20.11
ssh -l ucs-example\\jsmith 2001::1
```

- **ssh** {*UCSM-ip-address* | *UCSM-ipv6-address* | *UCSM-host-name*} **-lucs-***auth-domain**username*

```
ssh 192.0.20.11 -l ucs-example\\jsmith
ssh 2001::1 -l ucs-example\\jsmith
```

- **sshucs-***auth-domain**username*@{*UCSM-ip-address*|*UCSM-ipv6-address*}

```
ssh ucs-ldap23\\jsmith@192.0.20.11
ssh ucs-ldap23\\jsmith@2001::1
```

텔넷을 사용하는 Linux 터미널에서 다음 구문을 사용합니다.

참고 텔넷은 기본적으로 비활성화되어 있습니다. 텔넷 활성화에 대한 지침은 텔넷 구성, 105 페이지의 내용을 참조하십시오.

- **telnet ucs-***UCSM-host-name* **ucs-***auth-domain\username*

```
telnet ucs-qa-10
login: ucs-ldap23\blradmin
```

- **telnetucs-**{*UCSM-ip-address*|*UCSM-ipv6-address*}*ucs-auth-domain\username*

```
telnet 10.106.19.12 2052
ucs-qa-10-A login: ucs-ldap23\blradmin
```

Putty 클라이언트에서 다음 구문을 사용합니다.

- Login as: **ucs-***auth-domain\username*

```
Login as: ucs-example\jsmith
```

참고 기본 인증이 로컬로 설정되고 콘솔 인증이 LDAP으로 설정된 경우 ucs-local\admin을 사용하여 Putty 클라이언트에서 Fabric Interconnect에 로그인할 수 있습니다. 여기서 admin은 로컬 어카운트의 이름입니다.

4 장

ASA용 라이선스 관리

Cisco Smart Software Licensing을 사용하면 중앙 집중식으로 라이선스 풀을 구매하고 관리할 수 있습니다. 각 유닛의 라이선스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. Smart Software Licensing은 라이선스 사용량과 수요를 한 번에 확인하게 해줍니다.

참고

이 섹션은 Firepower 4100/9300 섀시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 컨피그레이션 가이드를 참조하십시오.

Smart Software Licensing 정보

이 섹션에서는 Smart Software Licensing이 적용되는 방법에 관해 설명합니다.

참고 이 섹션은 Firepower 4100/9300 섀시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 컨피그레이션 가이드를 참조하십시오.

ASA용 Smart Software Licensing

Firepower 4100/9300 섀시의 ASA 애플리케이션의 경우, Smart Software Licensing 컨피그레이션은 Firepower 4100/9300 섀시 수퍼바이저와 애플리케이션으로 나뉩니다.

- Firepower 4100/9300 섀시 — 수퍼바이저에 모든 Smart Software Licensing 인프라를 구성합니다(License Authority와 통신하는 데 필요한 파라미터 포함). Firepower 4100/9300 섀시 자체는 작동하기 위한 라이선스가 필요하지 않습니다.

  참고 섀시 간 클러스터링을 위해 클러스터에서 각 섀시에 동일한 Smart Licensing 방법을 활성화해야 합니다.

- ASA 애플리케이션 — 애플리케이션에서 모든 라이선스 엔타이틀먼트를 구성합니다.

Smart Software Manager 및 어카운트

디바이스 라이선스를 1개 이상 구매한 경우, Cisco Smart Software Manager에서 라이선스를 관리할 수 있습니다.

https://software.cisco.com/#module/SmartLicensing

Smart Software Manager를 활용하면 조직에서 사용할 마스터 어카운트를 만들 수 있습니다.

참고 아직 어카운트가 없는 경우 새 어카운트 설정 링크를 클릭합니다. Smart Software Manager를 활용하면 조직에서 사용할 마스터 어카운트를 만들 수 있습니다.

기본적으로 라이선스는 마스터 어카운트의 기본 가상 어카운트에 할당됩니다. 어카운트 관리자로서 선택적으로 추가 가상 어카운트를 생성할 수 있습니다. 예를 들어, 지역, 부서 또는 자회사에 대해 어카운트를 생성할 수 있습니다. 여러 가상 어카운트를 활용하면 많은 라이선스 및 디바이스를 더 쉽게 관리할 수 있습니다.

오프라인 관리

디바이스에서 인터넷에 액세스할 수 없으며 License Authority에 등록할 수 없는 경우, 오프라인 라이선싱을 구성할 수 있습니다.

영구 라이선스 예약

보안상의 이유로 디바이스에서 인터넷에 액세스할 수 없는 경우 선택적으로 각 ASA에 대해 영구 라이선스를 요청할 수 있습니다. 영구 라이선스에는 License Authority에 대한 주기적인 액세스가 필요하지 않습니다. PAK 라이선스와 같이 라이선스를 구매하고 ASA용 라이선스 키를 설치합니다. 그러나 PAK 라이선스와 달리 Smart Software Manager를 사용하여 라이선스를 얻고 관리합니다. 일반 Smart Licensing 모드와 영구 라이선스 예약 모드 간에 쉽게 전환할 수 있습니다.

모든 기능을 활성화하는 라이선스를 얻을 수 있습니다(최대 보안 상황 및 캐리어 라이선스가 있는 표준 계층). 라이선스는 Firepower 4100/9300 섀시에서 관리되지만 ASA가 사용을 허용하도록 ASA 컨피그레이션에서 엔타이틀먼트를 요청해야 합니다.

Satellite 서버

보안상의 이유로 디바이스에서 인터넷에 액세스할 수 없는 경우 선택적으로 로컬 Smart Software Manager Satellite 서버를 VM(Virtual Machine)으로 설치할 수 있습니다. Satellite에서는 Smart Software Manager 기능의 하위 집합을 제공하며, 모든 로컬 디바이스에 대한 필수 라이선싱 서비스를 제공할 수 있도록 허용합니다. Satellite의 경우에만 라이선스 사용량을 동기화하려면 기본 License Authority에 주기적으로 연결해야 합니다. 예약하여 동기화하거나 수동으로 동기화할 수 있습니다.

Satellite 애플리케이션을 다운로드 및 구축하고 나면 인터넷을 사용하여 Cisco SSM에 데이터를 전송하지 않고도 다음 기능을 수행할 수 있습니다.

- 라이선스 활성화 또는 등록
- 회사의 라이선스 확인
- 회사 엔터티 간 라이선스 양도

자세한 내용은 Smart Account Manager Satellite의 Smart Software Manager Satellite 설치 및 컨피그레이션 가이드를 참조하십시오.

가상 어카운트별로 관리되는 라이선스 및 디바이스

라이선스 및 디바이스는 가상 어카운트별로 관리됩니다. 가상 어카운트의 디바이스에서만 해당 어카운트에 할당된 라이선스를 사용할 수 있습니다. 추가 라이선스가 필요할 경우 다른 가상 어카운트의 미사용 라이선스를 이전할 수 있습니다. 또한, 가상 어카운트 간에 디바이스를 이전할 수도 있습니다.

Firepower 4100/9300 섀시만 디바이스로 등록되며, 섀시의 ASA 애플리케이션은 고유한 라이선스를 요청합니다. 예를 들어, 보안 모듈이 3개 있는 Firepower 9300 섀시의 경우 섀시는 디바이스 1개로 간주되지만 모듈은 별도의 라이선스를 3개 사용합니다.

평가판 라이선스

Firepower 4100/9300 섀시는 두 가지 유형의 평가판 라이선스를 지원합니다.

- 섀시 레벨 평가 모드 — Firepower 4100/9300 섀시가 Licensing Authority에 등록되기 전에 평가 모드로 90일(총 사용량) 동안 작동됩니다. ASA는 이 모드에서 특정 엔타이틀먼트를 요청할 수 없으며, 기본 엔타이틀먼트만 활성화됩니다. 이 기간이 끝나면 Firepower 4100/9300 섀시는 컴플라이언스 위반 상태가 됩니다.
- 엔타이틀먼트 기반 평가 모드 — Firepower 4100/9300 섀시가 Licensing Authority에 등록되고 나면 ASA에 할당 가능한 한시적인 평가판 라이선스를 얻을 수 있습니다. ASA에서 평소와 같이 엔타이틀먼트를 요청하십시오. 한시적인 라이선스가 만료되면 한시적인 라이선스를 갱신하거나 영구 라이선스를 얻어야 합니다.

참고 강력한 암호화(3DES/AES)를 위한 평가판 라이선스는 받을 수 없습니다. 즉, 영구 라이선스만 이 엔타이틀먼트를 지원합니다.

Smart Software Manager 통신

이 섹션에서는 디바이스가 Smart Software Manager와 통신하는 방법을 설명합니다.

디바이스 등록 및 토큰

각 가상 어카운트에서 등록 토큰을 생성할 수 있습니다. 이 토큰은 기본적으로 30일간 유효합니다. 각 섀시를 구축할 때 또는 기존 섀시를 등록할 때 이 토큰 ID와 엔타이틀먼트 레벨을 입력합니다. 기존 토큰이 만료되면 새 토큰을 생성할 수 있습니다.

구축 후 시작할 때 또는 기존 섀시에서 이러한 파라미터를 수동으로 구성한 후에 섀시가 Cisco License Authority에 등록됩니다. 섀시를 토큰과 함께 등록하면 License Authority는 섀시와 License Authority 간의 통신을 위한 ID 인증서를 발급합니다. 이 인증서는 6개월마다 갱신되지만 1년간 유효합니다.

License Authority와의 주기적인 통신

디바이스는 30일마다 License Authority와 통신합니다. Smart Software Manager에서 변경하는 경우 디바이스에서 권한 부여를 새로고침하여 변경 사항을 즉시 적용할 수 있습니다. 또는 디바이스에서 예정대로 통신할 때까지 기다릴 수 있습니다.

선택 사항으로 HTTP 프록시를 구성할 수 있습니다.

Firepower 4100/9300 섀시는 적어도 90일마다 직접 또는 HTTP 프록시를 통해 인터넷에 액세스할 수 있어야 합니다. 일반 라이선스 통신은 30일마다 이루어지지만, 유예 기간이 있으므로 디바이스는 최대 90일간 Call Home 없이 작동할 수 있습니다. 유예 기간이 지나면 License Authority와 통신해야 합니다. 그렇지 않으면 특별 라이선스가 필요한 기능의 컨피그레이션을 변경할 수 없지만 작업은 달리 영향을 받지 않습니다.

규정 위반 상태

디바이스는 다음과 같은 상황에서 컴플라이언스 위반 상태가 될 수 있습니다.

- 과다 사용 — 디바이스에서 사용 불가한 라이선스를 사용하는 경우
- 라이선스 만료—한시적인 라이선스가 만료된 경우
- 통신 부재 — 디바이스에서 권한 재부여를 위해 Licensing Authority에 연결하지 못하는 경우

어카운트가 컴플라이언스 위반 상태인지 또는 컴플라이언스 위반 상태에 근접한지를 확인하려면, Firepower 4100/9300 섀시에서 현재 사용 중인 엔타이틀먼트와 Smart Account의 엔타이틀먼트를 비교해야 합니다.

컴플라이언스 위반 상태인 경우 특별 라이선스가 필요한 기능의 컨피그레이션을 변경할 수 없지만 작업은 달리 영향을 받지 않습니다. 예를 들어, 표준 라이선스 제한을 통한 기존 상황을 계속 실행할 수 있으며 이러한 컨피그레이션을 수정할 수 있지만 새로운 상황을 추가할 수는 없습니다.

Smart Call Home 인프라

기본적으로, Smart Call Home 프로파일은 Licensing Authority의 URL을 지정하는 FXOS 컨피그레이션에 있습니다. 이 프로파일은 제거할 수 없습니다. 라이선스 프로파일의 유일한 컨피그레이션 옵션은 License Authority의 대상 주소 URL입니다. Cisco TAC에서 지시하지 않는 한 License Authority URL을 변경해서는 안 됩니다.

Smart Software Licensing 사전 요구 사항

- 이 장은 Firepower 4100/9300 섀시에 있는 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 컨피그레이션 가이드를 참조하십시오.
- Cisco Smart Software Manager에서 마스터 어카운트를 만듭니다.

  https://software.cisco.com/#module/SmartLicensing

  아직 어카운트가 없는 경우 새 어카운트 설정 링크를 클릭합니다. Smart Software Manager를 활용하면 조직에서 사용할 마스터 어카운트를 만들 수 있습니다.
- Cisco Commerce Workspace에서 라이선스를 1개 이상 구매합니다. 홈페이지의 **Find Products and Solutions**(제품 및 솔루션 찾기) 검색 필드에서 플랫폼을 검색합니다. 일부 라이선스는 무료이지만 Smart Software Licensing 어카운트에 추가해야 합니다.
- 섀시에서 Licensing Autority와 통신할 수 있도록 섀시에서 인터넷 액세스 또는 HTTP 프록시 액세스를 보장합니다.
- 섀시에서 Licensing Autority의 이름을 확인할 수 있도록 DNS 서버를 구성합니다.
- 섀시의 시간을 설정합니다.

- ASA 라이선싱 엔타이틀먼트를 구성하기 전에 Firepower 4100/9300 섀시에서 Smart Software Licensing 인프라를 구성합니다.

Smart Software Licensing 지침

장애 조치 및 클러스터링을 위한 **ASA** 지침

각 Firepower 4100/9300 섀시를 License Authority 또는 Satellite 서버에 등록해야 합니다. 보조 유닛에는 추가 비용이 없습니다. 영구 라이선스 예약의 경우, 각 섀시의 개별 라이선스를 구매해야 합니다.

Smart Software Licensing의 기본값

Firepower 4100/9300 섀시 기본 컨피그레이션에는 Licensing Authority의 URL을 지정하는 Smart Call Home 프로필 "SLProf"가 포함되어 있습니다.

```
scope monitoring
  scope callhome
    scope profile SLProf
      scope destination SLDest
        set address https://tools.cisco.com/its/service/oddce/services/DDCEService
```

일반 Smart Software Licensing 구성

Cisco License Authority와 통신하기 위해 선택적으로 HTTP 프록시를 구성할 수 있습니다. License Authority에 등록하려면 Smart Software License 어카운트에서 얻은 Firepower 4100/9300 섀시에 등록 토큰 ID를 입력해야 합니다.

절차

단계 **1** (선택 사항) HTTP 프록시 구성, 22 페이지.

단계 **2** License Authority에 Firepower Security Appliance 등록, 24 페이지.

(선택 사항) HTTP 프록시 구성

네트워크에서 인터넷 액세스를 위해 HTTP 프록시를 사용하는 경우, Smart Software Licensing에 대한 프록시 주소를 구성해야 합니다. 이 프록시는 일반적으로 Smart Call Home에도 사용됩니다.

참고 인증을 사용하는 HTTP 프록시는 지원되지 않습니다.

절차

단계 **1** HTTP 프록시를 활성화합니다.
scope monitoring scope callhome set http-proxy-server-enable on

예제:

```
scope monitoring
  scope call-home
    set http-proxy-server-enable on
```

단계 **2** 프록시 URL을 설정합니다.
set http-proxy-server-url *url*

여기서 *url*은 프록시 서버의 http 또는 https 주소입니다.

예제:

```
set http-proxy-server-url https://10.1.1.1
```

단계 **3** 포트를 설정합니다.
set http-proxy-server-port *port*

예제:

```
set http-proxy-server-port 443
```

단계 **4** 버퍼를 커밋합니다.
commit-buffer

(선택 사항) Call Home URL 삭제

다음 절차를 사용하여 이전에 구성한 Call Home URL을 삭제합니다.

절차

단계 **1** 모니터링 범위를 입력합니다.
scope monitoring

단계 **2** callhome 범위를 입력합니다.
scope callhome

단계 **3** SLProfile을 찾습니다.
scope profile SLProfile

단계 **4** 대상을 표시합니다.
show destination

예제:

```
SLDest https https://tools.cisco.com/its/oddce/services/DDCEService
```

단계 **5** URL을 삭제합니다.
delete destination SLDest

단계 **6** 버퍼를 커밋합니다.
commit-buffer

License Authority에 Firepower Security Appliance 등록

Firepower 4100/9300 섀시를 등록하면 License Authority에서는 Firepower 4100/9300 섀시와 License Authority 간의 통신을 위해 ID 인증서를 발급합니다. 또한, Firepower 4100/9300 섀시를 적절한 가상 어카운트에 할당합니다. 일반적으로 이 절차는 1회 수행됩니다. 그러나 예를 들어 나중에 통신 문제 때문에 ID 인증서가 만료되는 경우 Firepower 4100/9300 섀시를 다시 등록해야 할 수 있습니다.

절차

단계 **1** Smart Software Manager 또는 Smart Software Manager Satellite에서 이 Firepower 4100/9300 섀시를 추가할 가상 어카운트의 등록 토큰을 요청 및 복사합니다.
Smart Software Manager Satellite를 사용하여 등록 토큰을 요청하는 방법에 대한 자세한 내용은 Cisco Smart Software Manager Satellite 사용자 가이드(http://www.cisco.com/web/software/286285517/138897/Smart_Software_Manager_satellite_4.1.0_User_Guide.pdf)를 참조하십시오.

단계 **2** Firepower 4100/9300 섀시에 등록 토큰을 입력합니다.
scope license register idtoken *id-token*

예제:

```
scope license
  register idtoken ZGFmNWM5NjgtYmNjYS00ZWI3L
WE3NGItMWJkOGExZjIxNGQ0LTE0NjI2NDYx%0AMDIzNT
V8N3R0dXM1Z0NjWkdpR214eFZhMldBOS9CVnNEYnVKM1
g3R3dvemRD%0AY29NQT0%3D%0A
```

단계 **3** 이후에 디바이스의 등록을 취소하려면 다음을 입력합니다.
deregister

Firepower 4100/9300 섀시의 등록을 취소하면 어카운트에서 해당 디바이스가 제거됩니다. 디바이스의 모든 라이선스 엔타이틀먼트 및 인증서가 제거됩니다. 새 Firepower 4100/9300 섀시의 라이선스를 확보하기 위해 등록을 취소하는 경우가 있습니다. 또는 Smart Software Manager에서 해당 디바이스를 제거할 수 있습니다.

단계 **4** 모든 보안 모듈에서 ID 인증서를 갱신하고 엔타이틀먼트를 업데이트하려면 다음을 입력합니다.
scope licdebug renew

기본적으로 ID 인증서는 6개월마다 자동으로 갱신되며, 라이선스 엔타이틀먼트는 30일마다 갱신됩니다. 예를 들어 인터넷 액세스 기간이 제한된 경우 또는 Smart Software Manager에서 라이선스를 변경한 경우, 이러한 항목 중 하나에 대한 등록을 수동으로 갱신할 수 있습니다.

Smart License Satellite Server 구성 - Firepower 4100/9300 섀시

다음 절차에서는 Smart License Satellite Server를 사용하도록 Firepower 4100/9300 섀시를 구성하는 방법을 보여줍니다.

시작하기 전에

- Smart Software Licensing 사전 요구 사항, 21 페이지에 나열된 모든 사전 요구 사항을 완료합니다.
- Cisco.com에서 Smart License Satellite OVA 파일을 다운로드하고 VMwareESXi 서버에 이 파일을 설치 및 구성합니다. 자세한 내용은 Smart Software Manager Satellite 설치 가이드를 참조하십시오.
- 인증서 체인이 아직 없는 경우, 다음 절차를 사용하여 하나를 요청합니다.
  - 키 링을 생성합니다(키 링 생성, 114 페이지).
  - 키 링에 대한 인증서 요청을 생성합니다(기본 옵션을 사용하여 키 링에 대한 인증서 요청 생성, 116 페이지).
  - 이 인증서 요청을 Trust Anchor 또는 인증 기관(CA)에 전송하여 키 링에 대한 인증서 체인을 얻습니다.

자세한 내용은 인증서, 키 링 및 트러스트 포인트, 113 페이지를 참조하십시오.

절차

단계 **1** Callhome 대상으로 Satellite 서버를 설정합니다.

scopemonitoring

scopecall-home

scopeprofileSLProfile

scopedestinationSLDest

setaddresshttps://*ip_address***/Transportgateway/services/DeviceRequestHandler**

단계 **2** 새 트러스트 포인트를 생성합니다.

a) 보안 모드를 시작합니다.

scopesecurity

b) 트러스트 포인트를 생성하고 이름을 지정합니다.
createtrustpoint *trustpoint_name*

c) 트러스트 포인트에 대한 인증서 정보를 지정합니다. 참고: 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.
setcertchain *certchain*

certchain 변수의 경우, 이 절차의 인증서 생성 사전 요구 사항을 수행하여 얻은 인증서 체인 정보를 사용합니다.

명령에 인증서 정보를 지정하지 않은 경우, 루트 CA(인증 기관)에 인증 경로를 정의하는 트러스트 포인트를 목록 또는 인증서를 입력하라는 프롬프트가 표시됩니다. 해당 정보를 입력한 후 다음 행에 **ENDOFBUF**를 입력하여 완료합니다.

d) 컨피그레이션을 커밋합니다.
commit-buffer

예제:

```
firepower-chassis# scope security
firepower-chassis /security # create trustpoint tPoint10
firepower-chassis /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMDCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKONDl
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> jtcEMyZ+f7+3yh421ido3nO4MIGeBgNVHSMEgZYwgZOAFLlNjtcEMyZ+f7+3yh42
> 1ido3nO4oXikdjB0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAcT
> C1NhbnRhIENsYXJhMRswGQYDVQQKExJOdW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0VuZ2luZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAQAwDAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGQqXc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8copP1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIZJwK5+N3x/kfa2EHU6id1avt
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
firepower-chassis /security/trustpoint* # commit-buffer
firepower-chassis /security/trustpoint #
```

단계 **3** License Authority에 Firepower 4100/9300 섀시를 등록합니다(License Authority에 Firepower Security Appliance 등록, 24 페이지 참조). Smart License Manager Satellite에서 등록 토큰을 요청하고 복사해야 합니다.

영구 라이선스 예약 구성

영구 라이선스를 Firepower 4100/9300 섀시에 할당할 수 있습니다. 이 범용 예약을 통해 디바이스에서 무제한으로 모든 엔타이틀먼트를 사용하도록 허용할 수 있습니다.

참고 시작하기 전에 Smart Software Manager에서 사용할 수 있도록 영구 라이선스를 구매해야 합니다. 모든 어카운트가 영구 라이선스 예약에 대해 승인되지는 않습니다. 이 기능을 구성하려고 시도하기 전에 이 기능에 대해 Cisco의 승인을 받아야 합니다.

영구 라이선스 설치

다음 절차에서는 Firepower 4100/9300 섀시에 영구 라이선스를 할당하는 방법을 보여줍니다.

절차

단계 **1** FXOS CLI에서 라이선스 예약을 활성화합니다.
scope license

enable reservation

단계 **2** 라이선스 예약의 범위를 지정합니다.
scope license

scope reservation

단계 **3** 예약 요청 코드를 생성합니다.
request universal

show license resvcode

단계 **4** Cisco Smart Software Manager 포털에서 Smart Software Manager 인벤토리 화면으로 이동하고 **Licenses**(라이선스) 탭을 클릭합니다.
https://software.cisco.com/#SmartLicensing-Inventory

Licenses(라이선스) 탭은 어카운트와 관련된 모든 기존 라이선스(일반 및 영구 라이선스)를 표시합니다.

단계 **5** **License Reservation**(라이선스 예약)을 클릭하고 생성된 예약 요청 코드를 상자에 입력합니다.

단계 **6** **Reserve License**(라이선스 예약)를 클릭합니다.
Smart Software Manager는 권한 부여 코드를 생성합니다. 코드를 다운로드하거나 클립보드에 복사할 수 있습니다. 이 시점에서는 Smart Software Manager에 따라 라이선스가 사용됩니다.

License Reservation(라이선스 예약) 버튼이 보이지 않는 경우, 어카운트에 영구 라이선스 예약 권한이 없음을 의미합니다. 이 경우 영구 라이선스 예약을 비활성화하고 일반 smart license 명령을 다시 입력해야 합니다.

단계 **7** FXOS CLI에서 권한 부여 코드를 입력합니다.
license smart reservation install code

이제 Firepower 4100/9300 섀시에 PLR로 완전히 라이선스가 부여되었습니다.

단계 **8** ASA 논리적 디바이스에서 기능 엔타이틀먼트를 활성화합니다. 엔타이틀먼트를 활성화하려면 ASA 라이선싱 장을 참조하십시오.

(선택 사항) 영구 라이선스 반환

더 이상 영구 라이선스가 필요하지 않은 경우, 이 절차를 수행하여 해당 라이선스를 Smart Software Manager에 공식적으로 반환해야 합니다. 모든 단계를 따르지 않는 경우, 라이선스는 사용 중인 상태로 유지되며 다른 곳에서 사용할 수 없습니다.

절차

단계 **1** FXOS CLI에서 반환 코드를 생성합니다.
license smart reservation return

Firepower 4100/9300 섀시의 라이선스가 즉시 해제되고 평가 상태로 이동됩니다.

단계 **2** Smart Software Manager에서 FXOS 인스턴스를 찾을 수 있도록 FXOS UDI(universal device identifier)를 확인합니다.
show license udi

단계 **3** Smart Software Manager 인벤토리 화면으로 이동하고 **Product Instances**(제품 인스턴스) 탭을 클릭합니다.
https://software.cisco.com/#SmartLicensing-Inventory

단계 **4** UDI(universal device identifier)를 사용하여 Firepower 4100/9300 섀시를 검색합니다.

단계 **5** **Actions**(작업) > **Remove**(제거)를 선택하고 생성된 반환 코드를 상자에 입력합니다.

단계 **6** **Remove Product Instance**(제품 인스턴스 제거)를 클릭합니다.
영구 라이선스가 사용 가능한 풀로 반환됩니다.

Smart Software Licensing 모니터링

라이선스 상태를 보려면 다음 명령을 참조하십시오.

- **show license all**

  Smart Software Licensing 상태, 스마트 에이전트 버전, UDI 정보, 스마트 에이전트 상태, 글로벌 컴플라이언스 상태, 엔타이틀먼트 상태, 라이선싱 인증서 정보 및 스마트 에이전트 작업 일정을 표시합니다.

- **show license status**
- **show license techsupport**

Smart Software Licensing 기록

기능 이름	플랫폼 릴리스	설명
Firepower 4100/9300 섀시용 Cisco Smart Software Licensing	1.1(1)	Smart Software Licensing을 사용하면 라이선스 풀을 구입하고 관리할 수 있습니다. 스마트 라이선스는 특정 시리얼 번호에 연결되어 있지 않습니다. 각 유닛의 라이선스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. Smart Software Licensing은 라이선스 사용량과 수요를 한 번에 확인하게 해줍니다. Smart Software Licensing 컨피그레이션은 Firepower 4100/9300 섀시 수퍼바이저와 보안 모듈로 나뉩니다. 추가된 명령: **deregister, register idtoken, renew, scope callhome, scope destination, scope licdebug, scope license, scope monitoring, scope profile, set address, set http-proxy-server-enable on, set http-proxy-server-url, set http-proxy-server-port, show license all, show license status, show license techsupport**

5 장

사용자 관리

사용자 어카운트

사용자 어카운트는 시스템에 액세스하는 데 사용됩니다. 로컬 사용자 어카운트는 최대 48개 구성할 수 있습니다. 각 사용자 어카운트에는 고유한 사용자 이름과 비밀번호가 있어야 합니다.

관리자 어카운트

관리자 어카운트는 기본 사용자 어카운트이며 수정 또는 삭제할 수 없습니다. 이 어카운트는 시스템 관리자 또는 슈퍼 사용자(superuser) 어카운트이며 전체 권한을 갖습니다. 관리자 어카운트에는 기본 비밀번호가 할당되지 않으므로 초기 시스템 설정 시 비밀번호를 선택해야 합니다.

관리자 어카운트는 항상 활성 상태이며 만료되지 않습니다. 관리자 어카운트는 비활성 상태로 구성할 수 없습니다.

로컬 인증 사용자 어카운트

로컬 인증 사용자 어카운트는 섀시를 통해 직접 인증되며, 관리자 또는 AAA 권한을 보유한 사용자에 의해 활성화 또는 비활성화될 수 있습니다. 로컬 사용자 어카운트를 비활성화한 경우, 사용자는 로그인할 수 없습니다. 비활성화된 로컬 사용자 어카운트에 대한 컨피그레이션 세부사항은 데이터베이스에서 삭제되지 않습니다. 비활성화된 로컬 사용자 어카운트를 다시 활성화하면 해당 어카운트는 사용자 이름 및 비밀번호를 포함하여 기존 컨피그레이션으로 다시 활성화됩니다.

원격 인증 사용자 어카운트

원격 인증 사용자 어카운트는 LDAP, RADIUS 또는 TACACS+를 통해 인증되는 모든 사용자 어카운트를 가리킵니다.

사용자가 로컬 사용자 어카운트와 원격 사용자 어카운트를 동시에 유지할 경우 로컬 사용자 어카운트에 정의된 역할이 원격 사용자 어카운트의 역할을 재정의합니다.

원격 인증 지침에 대한 자세한 내용과 원격 인증 제공자를 구성 및 삭제하는 방법을 확인하려면 다음 항목을 참조하십시오.

- 원격 인증에 관한 지침, 34 페이지
- LDAP 제공자 구성, 126 페이지
- RADIUS 제공자 구성, 130 페이지
- TACACS+ 제공자 구성, 132 페이지

사용자 어카운트 만료

사전 정의된 시간에 만료하도록 사용자 어카운트를 구성할 수 있습니다. 만료 시간이 되면 사용자 어카운트가 비활성화됩니다.

기본적으로 사용자 어카운트는 만료되지 않습니다.

만료일이 있는 사용자 어카운트를 구성한 후에는 이 어카운트를 만료되지 않도록 재구성할 수 없습니다. 그러나 어카운트에 최신 만료일을 사용할 수 있도록 구성할 수는 있습니다.

사용자 이름 지침

사용자 이름은 Firepower Chassis Manager 및 FXOS CLI의 로그인 ID로도 사용됩니다. 사용자 어카운트에 로그인 ID를 할당할 때 다음 지침 및 제한 사항을 고려합니다.

- 로그인 ID는 1~32자이며 다음을 포함할 수 있습니다.
  - 모든 영문자
  - 모든 숫자
  - _(밑줄)
  - -(대시)
  - . (점)
- 로그인 ID는 고유해야 합니다.
- 로그인 ID는 영문자로 시작해야 합니다. 로그인 ID는 숫자 또는 특수 문자(예: 밑줄)로 시작할 수 없습니다.
- 로그인 ID는 대/소문자를 구분합니다.
- 전부 숫자로 된 로그인 ID는 만들 수 없습니다.
- 사용자 어카운트를 만든 후에는 로그인 ID를 변경할 수 없습니다. 사용자 어카운트를 삭제하고 새로 생성해야 합니다.

비밀번호 지침

로컬에서 인증되는 각 사용자 어카운트에는 비밀번호가 필요합니다. 관리자 또는 AAA 권한이 있는 사용자는 사용자 비밀번호에 대한 비밀번호 보안 강도를 확인하도록 시스템을 구성할 수 있습니다. 비밀번호 강도 확인이 활성화되면 각 사용자는 강력한 비밀번호를 사용해야 합니다.

각 사용자는 강력한 비밀번호를 사용하는 것이 좋습니다. 로컬로 인증된 사용자를 위해 비밀번호 보안 강도 확인을 활성화한 경우, Firepower eXtensible 운영 체제에서는 다음 요건을 충족하지 않는 비밀번호를 거부합니다.

- 최소 8자, 최대 80자를 포함해야 합니다.

참고 Common Criteria 요건을 준수하도록 시스템에서 최소 15자의 비밀번호 길이를 구성할 수도 있습니다. 자세한 내용은 최소 비밀번호 길이 확인 구성, 45 페이지를 참조하십시오.

- 하나 이상의 알파벳 대문자를 포함해야 합니다.
- 하나 이상의 알파벳 소문자를 포함해야 합니다.

- 하나 이상의 영숫자가 아닌(특수) 문자를 포함해야 합니다.
- aaabbb와 같이 한 문자가 3번 이상 연속적으로 반복되어서는 안 됩니다.
- 어떤 순서로도 3개의 연속되는 숫자 또는 문자(예: passwordABC 또는 password321)를 포함해서는 안 됩니다.
- 사용자 이름 또는 사용자 이름의 역순과 같아서는 안 됩니다.
- 비밀번호 사전 검사를 통과해야 합니다. 예를 들어, 비밀번호에 표준 사전 단어를 사용해서는 안 됩니다.
- $(달러 기호), ?(물음표), =(등호) 기호를 포함해서는 안 됩니다.
- 로컬 사용자 및 관리자 어카운트는 비어 있지 않아야 합니다.

원격 인증에 관한 지침

지원되는 원격 인증 서비스 중 하나에 대해 시스템이 구성되어 있는 경우 Firepower 4100/9300 섀시에서 시스템과 통신할 수 있도록 해당 서비스에 대한 제공자를 생성해야 합니다. 다음 지침은 사용자 권한 부여에 영향을 미칩니다.

원격 인증 서비스의 사용자 어카운트

사용자 어카운트는 로컬 Firepower 4100/9300 섀시에 또는 원격 인증 서버에 존재할 수 있습니다.

Firepower Chassis Manager 또는 FXOS CLI에서 원격 인증 서비스로 로그인한 사용자의 임시 세션을 볼 수 있습니다.

원격 인증 서비스의 사용자 역할

원격 인증 서버에 사용자 어카운트를 생성하는 경우 해당 어카운트는 사용자가 Firepower 4100/9300 섀시에서 작업하는 데 필요한 역할을 포함해야 하며 해당 역할의 이름이 FXOS에서 사용되는 이름과 일치해야 합니다. 역할 정책에 따라 사용자가 로그인하지 못하거나 읽기 전용 권한만 가질 수도 있습니다.

원격 인증 제공자의 사용자 속성

RADIUS 및 TACAS+ 컨피그레이션에서는 각 원격 인증 제공자(이를 통해 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인)의 Firepower 4100/9300 섀시에 대한 사용자 속성을 구성해야 합니다. 이 사용자 속성에는 각 사용자에게 할당된 역할 및 로캘이 저장됩니다.

사용자가 로그인하면 FXOS에서 다음을 수행합니다.

1. 원격 인증 서비스에 대해 쿼리합니다.
2. 사용자를 검증합니다.
3. 사용자가 검증된 경우 사용자에게 할당된 역할 및 로캘을 확인합니다.

다음 표에서는 FXOS에서 지원하는 원격 인증 제공자의 사용자 속성 요구 사항을 비교합니다.

인증 제공자	맞춤형 속성	스키마 확장	속성 **ID** 요구 사항
LDAP	선택 사항	다음 중 하나를 선택하여 수행할 수 있습니다. • LDAP 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 속성을 구성합니다. • LDAP 스키마를 확장하고 CiscoAVPair와 같은 고유한 이름으로 맞춤형 속성을 생성합니다.	Cisco LDAP 구현에서는 유니코드 형식의 속성이 필요합니다. CiscoAVPair 맞춤형 속성을 생성하려는 경우 속성 ID로 1.3.6.1.4.1.9.287247.1을 사용합니다. 샘플 OID는 다음 섹션에서 제공됩니다.
RADIUS	선택 사항	다음 중 하나를 선택하여 수행할 수 있습니다. • RADIUS 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 속성을 사용합니다. • RADIUS 스키마를 확장하고 cisco-avpair와 같은 고유한 이름으로 맞춤형 속성을 생성합니다.	Cisco RADIUS 구현의 벤더 ID는 009, 속성의 벤더 ID는 001입니다. 다음 구문의 예에서는 cisco-avpair 속성을 생성하려는 경우 여러 사용자 역할 및 로케일을 지정하는 방법을 보여줍니다. `shell:roles="admin,aaa" shell:locales="L1,abc".` 여러 값을 구분하는 기호로 쉼표 ","를 사용합니다.

인증 제공자	맞춤형 속성	스키마 확장	속성 **ID** 요구 사항
TACAS	필수	스키마를 확장하고 cisco-av-pair라는 이름으로 맞춤형 속성을 생성해야 합니다.	cisco-av-pair 이름은 TACACS+ 제공자에 대한 속성 ID를 제공하는 문자열입니다. 다음 구문의 예에서는 cisco-av-pair 속성을 생성할 경우 여러 사용자 역할 및 로케일을 지정하는 방법을 보여줍니다. cisco-av-pair=shell:roles="admin aaa" shell:locales*"L1 abc". cisco-av-pair 속성 구문에 별표(*)를 사용하면 로케일에 선택 사항 플래그를 지정합니다. 그러면 동일한 권한 부여 프로필을 사용하는 다른 Cisco 디바이스의 인증이 실패하지 않습니다. 여러 값을 구분하는 기호로 공백을 사용합니다.

LDAP 사용자 속성에 대한 샘플 **OID**

다음은 맞춤형 CiscoAVPair 속성에 대한 샘플 OID입니다.

```
CN=CiscoAVPair,CN=Schema,
CN=Configuration,CN=X
objectClass: top
objectClass: attributeSchema
cn: CiscoAVPair
distinguishedName: CN=CiscoAVPair,CN=Schema,CN=Configuration,CN=X
instanceType: 0x4
uSNCreated: 26318654
attributeID: 1.3.6.1.4.1.9.287247.1
attributeSyntax: 2.5.5.12
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: CiscoAVPair
adminDescription: UCS User Authorization Field
oMSyntax: 64
lDAPDisplayName: CiscoAVPair
```

```
name: CiscoAVPair
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,CN=X
```

사용자 역할

시스템에는 다음과 같은 사용자 역할이 있습니다.

관리자

전체 시스템에 대한 완전한 읽기 및 쓰기 액세스 권한이 있습니다. 이 역할에는 기본적으로 기본 관리자 어카운트가 할당되며 이는 변경할 수 없습니다.

읽기 전용

시스템 컨피그레이션에 대한 읽기 전용 액세스 권한이 있습니다(시스템 상태를 수정할 권한은 없음).

작업

NTP 컨피그레이션, Smart Licensing용 Smart Call Home 컨피그레이션, 시스템 로그(syslog 서버 및 결함 포함)에 대한 읽기 및 쓰기 액세스 권한이 있습니다. 나머지 시스템에 대해서는 읽기 액세스 권한이 있습니다.

AAA 관리자

사용자, 역할, AAA 컨피그레이션에 대한 읽기 및 쓰기 액세스 권한이 있습니다. 나머지 시스템에 대해서는 읽기 액세스 권한이 있습니다.

로컬 인증 사용자에 대한 비밀번호 프로파일

비밀번호 프로파일에는 모든 로컬로 인증된 사용자에 대한 비밀번호 기록 및 비밀번호 변경 간격 속성이 포함되어 있습니다. 로컬로 인증된 각 사용자에 대해 다른 비밀번호 프로파일을 지정할 수 없습니다.

비밀번호 기록 수

비밀번호 기록 수를 사용하면 로컬에서 인증된 사용자가 동일한 비밀번호를 계속해서 재사용하는 것을 방지할 수 있습니다. 이 속성을 구성할 때, Firepower 섀시는 로컬에서 인증된 사용자가 이전에 사용한 비밀번호를 최대 15개까지 저장합니다. 비밀번호는 가장 최근 비밀번호부터 먼저 시간 순서대로 저장되어 기록 수 임계값에 도달했을 때 가장 오래된 비밀번호만 재사용할 수 있도록 합니다.

사용자는 먼저 비밀번호 기록 수에 구성되어 있는 개수만큼 비밀번호를 생성하고 사용해야 비밀번호를 재사용할 수 있습니다. 예를 들어, 비밀번호 기록 수를 8로 설정한 경우 로컬로 인증된 사용자는 9번째 비밀번호가 만료될 때까지 첫 번째 비밀번호를 재사용할 수 없습니다.

기본적으로 비밀번호 기록 수는 0으로 설정되어 있습니다. 이 값이 설정되면 기록 수를 비활성화하고 사용자가 언제든지 이전의 비밀번호를 재사용할 수 있습니다.

필요한 경우, 로컬로 인증된 사용자의 비밀번호 기록 수를 지우고 이전 비밀번호 재사용을 활성화할 수 있습니다.

비밀번호 변경 간격

비밀번호 변경 간격을 사용하면 로컬에서 인증된 사용자가 지정된 시간 내에 수행 가능한 비밀번호 변경 횟수를 제한할 수 있습니다. 다음 표에서는 비밀번호 변경 간격의 두 가지 컨피그레이션 옵션을 설명합니다.

간격 컨피그레이션	설명	예
비밀번호 변경 허용 안 함	이 옵션을 사용하면 비밀번호 변경 이후 지정된 시간 동안 로컬로 인증된 사용자 비밀번호의 변경이 허용되지 않습니다. 변경 불가 간격을 1~745시간으로 지정할 수 있습니다. 기본적으로 변경 불가 간격은 24시간입니다.	예를 들어 로컬에서 인증된 사용자가 비밀번호를 변경한 후 48시간 이내에 비밀번호를 변경하지 못하도록 하려면 다음을 설정합니다. • 변경 지속 간격 - 비활성화 • 변경 불가 간격 - 48
변경 간격 내에서 비밀번호 변경 허용	이 옵션은 로컬에서 인증된 사용자가 사전 정의된 간격 내에 비밀번호를 변경할 수 있는 최대 횟수를 지정합니다. 변경 간격을 1~745시간으로 지정하고 비밀번호 변경 최대 횟수를 0~10으로 지정할 수 있습니다. 기본적으로 로컬에서 인증된 사용자는 48시간 간격 내에 비밀번호를 최대 2번 변경하는 것이 허용됩니다.	예를 들어 로컬에서 인증된 사용자가 비밀번호를 변경한 후 24시간 이내에 비밀번호를 최대 1번 변경할 수 있도록 하려면 다음과 같이 설정합니다. • 변경 지속 간격 - 활성화 • 변경 횟수 - 1 • 변경 간격 - 24

기본 인증 서비스 선택

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scope security**

단계 **2** 기본 권한 부여 보안 모드를 시작합니다.
Firepower-chassis /security # **scopedefault-auth**

단계 **3** 기본 인증을 지정합니다.
Firepower-chassis /security/default-auth # **set realm** *auth-type*

여기서 *auth-type*은 다음 키워드 중 하나입니다.

- **ldap**— LDAP 인증 지정
- **local**— 로컬 인증 지정
- **none**— 로컬 사용자가 비밀번호를 지정하지 않고 로그온하도록 허용
- **radius**— RADIUS 인증 지정
- **tacacs**— TACACS+ 인증 지정

단계 **4** (선택 사항) 해당하는 경우, 연결된 제공자 그룹을 지정합니다.
Firepower-chassis /security/default-auth # **set auth-server-group** *auth-serv-group-name*

단계 **5** (선택 사항) 이 도메인에 있는 사용자에 대한 새로고침 요청 사이에 허용되는 최대 시간을 지정합니다.
Firepower-chassis /security/default-auth # **set refresh-period** *seconds*

0~600의 정수를 지정합니다. 기본값은 600초입니다.

이 시간 제한을 초과할 경우 FXOS에서는 웹 세션이 비활성화되는 것으로 간주하지만 세션을 종료하지는 않습니다.

단계 **6** (선택 사항) FXOS에서 웹 세션이 종료되었다고 간주하기 전 마지막 새로고침 요청 이후에 경과할 수 있는 최대 시간을 지정합니다.
Firepower-chassis /security/default-auth # **set session-timeout** *seconds*

0~600의 정수를 지정합니다. 기본값은 600초입니다.

참고 RADIUS 또는 TACACS+ 영역에 대한 2단계 인증을 설정하는 경우, 원격 사용자가 재인증을 자주 수행하지 않아도 되도록 **session-refresh** 및 **session-timeout** 간격을 늘려 설정해 보십시오.

단계 **7** (선택 사항) 영역에 대한 2단계 인증 방법을 설정합니다.
Firepower-chassis /security/default-auth # **set use-2-factor yes**

참고 2단계 인증은 RADIUS 및 TACACS+ 영역에만 적용됩니다.

단계 **8** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
commit-buffer

다음의 예에서는 기본 인증을 RADIUS에 설정하고, 기본 인증 제공자 그룹을 provider1로 설정하며, 2단계 인증을 활성화하고, 새로고침 간격을 300초(5분)로 설정하며, 세션 시간 초과 간격을 540초(9분)로 설정하고, 2단계 인증을 활성화합니다. 그런 다음 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope default-auth
Firepower-chassis /security/default-auth # set realm radius
Firepower-chassis /security/default-auth* # set auth-server-group provider1
Firepower-chassis /security/default-auth* # set use-2-factor yes
Firepower-chassis /security/default-auth* # set refresh-period 300
Firepower-chassis /security/default-auth* # set session-timeout 540
Firepower-chassis /security/default-auth* # commit-buffer
Firepower-chassis /security/default-auth #
```

세션 시간 초과 구성

FXOS CLI를 사용하여 Firepower 4100/9300 섀시가 사용자 세션을 닫기 전에 사용자 활동 없이 경과할 수 있는 시간을 지정할 수 있습니다. 콘솔 세션 및 HTTPS, SSH 및 텔넷 세션에 대해 서로 다른 설정을 구성할 수 있습니다.

최대 3,600초(60분)의 시간 초과 값을 설정할 수 있습니다. 기본값은 600초입니다. 이 설정을 비활성화하려면 세션 시간 초과 값을 0으로 설정합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scope security**

단계 **2** 기본 권한 부여 보안 모드를 시작합니다.
Firepower-chassis /security # **scopedefault-auth**

단계 **3** HTTPS, SSH 및 텔넷 세션에 대한 유휴 시간 초과를 설정합니다.
Firepower-chassis /security/default-auth # **set session-timeout** *seconds*

단계 **4** (선택 사항) 콘솔 세션에 대한 유휴 시간 초과를 설정합니다.
Firepower-chassis /security/default-auth # **set con-session-timeout** *seconds*

단계 **5** (선택 사항) 세션 및 절대 세션 시간 초과 설정을 확인합니다.
Firepower-chassis /security/default-auth # **show detail**

예제:

```
Default authentication:
 Admin Realm: Local
 Operational Realm: Local
 Web session refresh period(in secs): 600
 Session timeout(in secs) for web, ssh, telnet sessions: 600
 Absolute Session timeout(in secs) for web, ssh, telnet sessions: 3600
 Serial Console Session timeout(in secs): 600
 Serial Console Absolute Session timeout(in secs): 3600
 Admin Authentication server group:
 Operational Authentication server group:
 Use of 2nd factor: No
```

절대 세션 시간 초과 구성

Firepower 4100/9300 섀시에는 세션 사용과 상관없이 절대 세션 시간 초과 기간이 지나면 사용자 세션을 닫는 절대 세션 시간 초과 설정이 있습니다. 이 절대 시간 초과 기능은 시리얼 콘솔, SSH, HTTPS를 비롯한 모든 액세스 형식에서 전역적으로 적용됩니다.

시리얼 콘솔 세션의 절대 세션 시간 초과를 별도로 구성할 수 있습니다. 이렇게 하면 다른 형태의 액세스에 대한 시간 초과는 유지하면서 디버깅 요구 사항에 대한 직렬 콘솔 절대 세션 시간 초과를 비활성화할 수 있습니다.

절대 시간 초과 기본값은 3600초(60분)이며 FXOS CLI를 사용해 변경할 수 있습니다. 이 설정을 비활성화하려면 절대 세션 시간 초과 값을 0으로 설정합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scope security**

단계 **2** 기본 권한 부여 보안 모드를 시작합니다.
Firepower-chassis /security # **scopedefault-auth**

단계 **3** 절대 세션 시간 초과를 설정합니다.
Firepower-chassis /security/default-auth # **set absolute-session-timeout** *seconds*

단계 **4** (선택 사항) 개별 콘솔 절대 세션 시간 초과를 설정합니다.
Firepower-chassis /security/default-auth # **set con-absolute-session-timeout** *seconds*

단계 **5** (선택 사항) 세션 및 절대 세션 시간 초과 설정을 확인합니다.
Firepower-chassis /security/default-auth # **show detail**

예제:

```
Default authentication:
 Admin Realm: Local
 Operational Realm: Local
 Web session refresh period(in secs): 600
 Session timeout(in secs) for web, ssh, telnet sessions: 600
 Absolute Session timeout(in secs) for web, ssh, telnet sessions: 3600
 Serial Console Session timeout(in secs): 600
 Serial Console Absolute Session timeout(in secs): 3600
 Admin Authentication server group:
 Operational Authentication server group:
 Use of 2nd factor: No
```

원격 사용자에 대한 역할 정책 구성

기본적으로 LDAP, RADIUS 또는 TACACS 프로토콜을 사용하여 원격 서버에서 Firepower Chassis Manager 또는 FXOS CLI에 로그인하는 모든 사용자에게 읽기 전용 액세스 권한이 부여됩니다. 보안상의 이유로, 설정된 사용자 역할과 일치하는 사용자로 액세스를 제한하는 것이 바람직할 수 있습니다.

다음과 같이 원격 사용자에 대한 역할 정책을 구성할 수 있습니다.

assign-default-role

사용자가 로그인을 시도하는데 원격 인증 제공자가 사용자 역할에 인증 정보를 제공하지 않는 경우, 사용자는 읽기 전용 사용자 역할로 로그인할 수 있습니다.

이는 기본 행동입니다.

no-login

사용자가 로그인을 시도하는데 원격 인증 제공자가 사용자 역할에 인증 정보를 제공하지 않는 경우, 액세스가 거부됩니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scope security**

단계 **2** Firepower Chassis Manager 및 FXOS CLI에 대한 사용자 액세스가 사용자 역할을 기준으로 제한되어야 하는지 여부를 지정합니다.
Firepower-chassis /security # **set remote-user default-role** {**assign-default-role** | **no-login**}

단계 **3** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security # **commit-buffer**

다음 예에서는 원격 사용자에 대한 역할 정책을 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # set remote-user default-role no-login
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

로컬 인증 사용자의 비밀번호 보안 강도 확인 활성화

비밀번호 보안 강도 확인이 활성화된 경우에는 Firepower eXtensible 운영 체제에서 사용자가 강력한 비밀번호 지침을 따르지 않는 비밀번호를 선택하도록 허용하지 않습니다(비밀번호 지침, 33 페이지 참조).

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scopesecurity**

단계 **2** 비밀번호 보안 강도 확인을 활성화할지 또는 비활성화할지 지정합니다.
Firepower-chassis /security # **set enforce-strong-password** {**yes** | **no**}

다음 예에서는 비밀번호 보안 강도 확인을 활성화합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # set enforce-strong-password yes
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

최대 로그인 시도 횟수 설정

Firepower 4100/9300 섀시에서 지정된 시간 동안 사용자가 잠기기 전에 허용되는 최대 로그인 시도 실패 횟수를 구성할 수 있습니다. 설정된 최대 로그인 시도 횟수를 초과하는 경우 사용자는 시스템에서 잠깁니다. 사용자가 잠겼음을 나타내는 알림은 나타나지 않습니다. 이 경우 사용자는 로그인을 시도하기 전에 지정된 시간 동안 기다려야 합니다.

최대 로그인 시도 횟수를 구성하려면 다음 단계를 수행합니다.

참고

- 최대 로그인 시도 횟수를 초과하고 나면 시스템에서 모든 유형의 사용자 어카운트(관리자 포함)가 잠깁니다.
- 기본 최대 로그인 시도 실패 횟수는 0입니다. 최대 로그인 시도 횟수를 초과하고 나서 시스템에서 사용자가 잠기는 기본 시간은 15분(900초)입니다.
- 사용자의 잠금 상태를 확인하고 사용자의 잠금 상태를 지우는 단계에 대해서는 사용자 잠금 상태 보기 및 지우기, 44 페이지의 내용을 참조하십시오.

이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증 컴플라이언스, 63 페이지를 참조하십시오.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesystem

scopesecurity

단계 **2** 최대 로그인 시도 실패 횟수를 설정합니다.
setmax-login-attempts

max_login
max_login 값은 0~10의 정수입니다.

단계 **3** 최대 로그인 시도 횟수에 도달하고 나서 시스템에서 사용자가 잠긴 상태로 유지되어야 하는 시간(초 단위)을 지정합니다.
setuser-account-unlock-time

unlock_time

단계 **4** 컨피그레이션을 커밋합니다.
commit-buffer

사용자 잠금 상태 보기 및 지우기

관리자 사용자는 Maximum Number of Login Attempts(최대 로그인 시도 횟수) CLI 설정에 지정된 최대 실패 로그인 횟수를 초과한 후 Firepower 4100/9300 섀시에서 잠긴 사용자의 잠금 상태를 보고 지울 수 있습니다. 자세한 내용은 최대 로그인 시도 횟수 설정, 43 페이지를 참조하십시오.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesystem

scopesecurity

단계 **2** 문제가 있는 사용자의 사용자 정보(잠금 상태 포함)를 표시합니다.
Firepower-chassis /security # **show local-user** *user***detail**

예제**:**

```
Local User user(로컬 사용자):
 First Name(이름):
 Last Name(성):
 Email(이메일):
 Phone(전화):
 Expiration(만료일): Never(없음)
 Password(비밀번호):
 User lock status(사용자 잠금 상태): Locked(잠김)
 Account status(계정 상태): Active(활성)
 User Roles(사용자 역할):
 Name(이름): read-only(읽기 전용)
 User SSH public key(사용자 SSH 공개 키):
```

단계 **3** (선택 사항) 사용자의 잠금 상태를 지웁니다.
Firepower-chassis /security # **scope local-user** *user*

Firepower-chassis /security/local-user # **clear lock-status**

변경 간격 동안 최대 비밀번호 변경 횟수 구성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scopesecurity**

단계 **2** 비밀번호 프로파일 보안 모드를 시작합니다.
Firepower-chassis /security # **scope password-profile**

단계 **3** 지정된 시간 이내에 로컬 인증 사용자가 비밀번호를 변경할 수 있는 횟수를 제한합니다.
Firepower-chassis /security/password-profile # **set change-during-interval enable**

단계 **4** 로컬 인증 사용자가 변경 간격 동안 비밀번호를 변경할 수 있는 최대 횟수를 지정합니다.
Firepower-chassis /security/password-profile # **set change-count** *pass-change-num*

이 값은 0~10으로 선택할 수 있습니다.

단계 **5** **Change Count(**변경 횟수**)** 필드에 지정된 비밀번호 변경 횟수가 적용되는 최대 시간을 지정합니다.
Firepower-chassis /security/password-profile # **set change-interval** *num-of-hours*

이 값은 1~745시간으로 선택할 수 있습니다.

예를 들어, 이 필드가 48로 설정되고 **Change Count(**변경 횟수**)** 필드가 2로 설정된 경우 로컬로 인증된 사용자는 48시간 이내에 비밀번호를 최대 2번 변경할 수 있습니다.

단계 **6** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/password-profile # **commit-buffer**

다음 예에서는 해당 간격 동안 변경 옵션을 활성화하고, 변경 횟수를 5로 설정한 다음, 변경 간격을 72시간으로 설정하고, 트랜잭션을 커밋합니다.

```
Firepower-chassis # scope security
Firepower-chassis /security # scope password-profile
Firepower-chassis /security/password-profile # set change-during-interval enable
Firepower-chassis /security/password-profile* # set change-count 5
Firepower-chassis /security/password-profile* # set change-interval 72
Firepower-chassis /security/password-profile* # commit-buffer
Firepower-chassis /security/password-profile #
```

최소 비밀번호 길이 확인 구성

최소 비밀번호 길이 확인을 활성화하는 경우, 지정된 최소 문자 수를 지닌 비밀번호를 생성해야 합니다. 예를 들어 *min_length* 옵션이 15로 설정된 경우 15자 이상을 사용해 비밀번호를 만들어야 합니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 허용하는 숫자 중 하나입니다. 자세한 내용은 보안 인증 컴플라이언스, 63 페이지를 참조하십시오.

최소 비밀번호 길이 확인을 구성하려면 다음 단계를 수행합니다.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.

단계 **2** **scopesystem**
scopesecurity

단계 **3** 비밀번호 프로파일 보안 모드를 시작합니다.
scopepassword-profile

단계 **4** 최소 비밀번호 길이를 지정합니다.
setmin-password-length *min_length*

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

비밀번호에 대해 변경 안 함 간격 구성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scopesecurity**

단계 **2** 비밀번호 프로파일 보안 모드를 시작합니다.
Firepower-chassis /security # **scope password-profile**

단계 **3** 해당 간격 동안 변경 기능을 비활성화합니다.
Firepower-chassis /security/password-profile # **set change-during-interval disable**

단계 **4** 로컬 인증 사용자가 새로 생성된 비밀번호를 변경하기 전까지 기다려야 하는 최소 시간을 지정합니다.
Firepower-chassis /security/password-profile # **set no-change-interval** *min-num-hours*

이 값은 1~745시간으로 선택할 수 있습니다.

이 간격은 **Change During Interval(**해당 간격 동안 변경**)** 속성이 **Disable(**비활성화**)**로 설정되지 않은 경우 무시됩니다.

단계 **5** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/password-profile # **commit-buffer**

다음 예에서는 해당 간격 동안 변경 옵션을 비활성화하고, 변경 안 함 간격을 72시간으로 설정한 다음, 트랜잭션을 커밋합니다.

```
Firepower-chassis # scope security
Firepower-chassis /security # scope password-profile
```

```
Firepower-chassis /security/password-profile # set change-during-interval disable
Firepower-chassis /security/password-profile* # set no-change-interval 72
Firepower-chassis /security/password-profile* # commit-buffer
Firepower-chassis /security/password-profile #
```

비밀번호 기록 수 구성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scopesecurity**

단계 **2** 비밀번호 프로파일 보안 모드를 시작합니다.
Firepower-chassis /security # **scope password-profile**

단계 **3** 로컬 인증 사용자가 이전에 사용한 비밀번호를 재사용하기 전에 생성해야 하는 고유한 비밀번호 수를 지정합니다.
Firepower-chassis /security/password-profile # **set history-count** *num-of-passwords*

0 ~ 15의 어떤 값이든 가능합니다.

기본적으로 **History Count(**기록 수**)** 필드는 0으로 설정되어 있습니다. 즉, 기록 수가 비활성화되어 있으므로 이전에 사용한 비밀번호를 언제라도 재사용할 수 있습니다.

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/password-profile # **commit-buffer**

다음 예에서는 비밀번호 기록 수를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis # scope security
Firepower-chassis /security # scope password-profile
Firepower-chassis /security/password-profile # set history-count 5
Firepower-chassis /security/password-profile* # commit-buffer
Firepower-chassis /security/password-profile #
```

로컬 사용자 어카운트 생성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 사용자 어카운트를 생성합니다.
Firepower-chassis /security # **create local-user** *local-user-name*

여기서 *local-user-name*은 이 어카운트에 로그인할 때 사용되는 어카운트 이름입니다. 이 이름은 고유해야 하며 사용자 어카운트 이름에 대한 지침 및 제한 사항을 따라야 합니다(사용자 이름 지침, 33 페이지 참조).

사용자를 생성하고 나면 로그인 ID를 변경할 수 없습니다. 사용자 어카운트를 삭제하고 새로 생성해야 합니다.

단계 **3** 로컬 사용자 어카운트를 활성화할지 또는 비활성화할지 지정합니다.
Firepower-chassis /security/local-user # **set account-status** {**active**| **inactive**}

단계 **4** 사용자 어카운트의 비밀번호를 설정합니다.
Firepower-chassis /security/local-user # **set password**

비밀번호 입력: *password*

비밀번호 확인: *password*

비밀번호 보안 강도 확인을 활성화한 경우, 사용자의 비밀번호가 더욱 강력해지며 Firepower eXtensible 운영 체제는 다음 보안 강도 확인 요건을 충족하지 않는 비밀번호를 거부합니다(비밀번호 지침, 33 페이지 참조).

단계 **5** (선택 사항) 사용자의 이름을 지정합니다.
Firepower-chassis /security/local-user # **set firstname** *first-name*

단계 **6** (선택 사항) 사용자의 성을 지정합니다.
Firepower-chassis /security/local-user # **set lastname** *last-name*

단계 **7** (선택 사항) 사용자 어카운트가 만료되는 날짜를 지정합니다. *month* 인수는 월 이름의 처음 세 글자입니다.
Firepower-chassis /security/local-user # **set expiration** *month day-of-month year*

참고 만료일이 있는 사용자 어카운트를 구성한 후에는 이 어카운트를 만료되지 않도록 재구성할 수 없습니다. 그러나 어카운트에 최신 만료일을 사용할 수 있도록 구성할 수는 있습니다.

단계 **8** (선택 사항) 사용자의 이메일 주소를 지정합니다.
Firepower-chassis /security/local-user # **set email** *email-addr*

단계 **9** (선택 사항) 사용자 전화 번호를 지정합니다.
Firepower-chassis /security/local-user # **set phone** *phone-num*

단계 **10** (선택 사항) 비밀번호 없는 액세스에 사용되는 SSH 키를 지정합니다.
Firepower-chassis /security/local-user # **set sshkey** *ssh-key*

단계 **11** 모든 사용자에게 기본적으로 *read-only* 역할이 할당되며 이 역할은 제거할 수 없습니다. 사용자에게 할당할 각 추가 역할에 대해 다음을 입력합니다.
Firepower-chassis /security/local-user # **create role** *role-name*

여기서 *role-name*은 사용자 어카운트에 할당할 권한에 해당하는 역할입니다(사용자 역할, 37 페이지 참조).

참고 사용자 역할 및 권한 변경 사항은 다음에 사용자가 로그인한 이후에 적용됩니다. 사용자 어카운트에 새 역할을 할당하거나 사용자 어카운트에서 기존 역할을 제거할 때 사용자가 로그인하는 경우, 활성화된 세션이 이전 역할 및 권한을 사용하여 작업을 계속합니다.

단계 **12** 사용자로부터 할당된 역할을 제거하려면 다음을 입력합니다.

Firepower-chassis /security/local-user # **delete role** *role-name*

참고 모든 사용자에게 기본적으로 *read-only* 역할이 할당되며 이 역할은 제거할 수 없습니다.

단계 **13** 트랜잭션을 커밋합니다.
Firepower-chassis security/local-user # **commit-buffer**

다음 예에서는 kikipopo라는 이름의 사용자 어카운트를 생성하고, 이 사용자 어카운트를 활성화한 다음, 비밀번호를 foo12345로 설정하고, 관리 사용자 역할을 할당한 후, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create local-user kikipopo
Firepower-chassis /security/local-user* # set account-status active
Firepower-chassis /security/local-user* # set password
Enter a password:
Confirm the password:
Firepower-chassis /security/local-user* # create role admin
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

다음 예에서는 lincey라는 이름의 사용자 어카운트를 생성하고, 이 사용자 어카운트를 활성화한 다음, 비밀번호 없는 액세스에 대해 OpenSSH 키를 설정하고, aaa 및 작업 사용자 역할을 할당한 후, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create local-user lincey
Firepower-chassis /security/local-user* # set account-status active
Firepower-chassis /security/local-user* #  set sshkey "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAuo9VQ2CmWBI9/S1f30klCWjnV3lgdXMzO0WUl5iPw85lkdQqap+NFuNmHcb4K
iaQB8X/PDdmtlxQQcawclj+k8f4VcOelBxlsGk5luq5ls1ob1VOIEwcKEL/h5lrdbNlI8y3SS9I/gGiBZ9ARlop9LDpD
m8HPh2LOgyH7Ei1MI8="
Firepower-chassis /security/local-user* # create role aaa
Firepower-chassis /security/local-user* # create role operations
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

다음 예에서는 jforlenz라는 이름의 사용자 어카운트를 생성하고, 이 사용자 어카운트를 활성화한 다음, 비밀번호 없는 액세스에 대해 보안 SSH 키를 설정하고, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create local-user jforlenz
Firepower-chassis /security/local-user* # set account-status active
Firepower-chassis /security/local-user* #  set sshkey
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
User's SSH key:
> ---- BEGIN SSH2 PUBLIC KEY ----
>AAAAB3NzaC1yc2EAAAABIwAAAIEAuo9VQ2CmWBI9/S1f30klCWjnV3lgdXMzO0WUl5iPw8
>5lkdQqap+NFuNmHcb4KiaQB8X/PDdmtlxQQcawclj+k8f4VcOelBxlsGk5luq5ls1ob1VO
>IEwcKEL/h5lrdbNlI8y3SS9I/gGiBZ9ARlop9LDpDm8HPh2LOgyH7Ei1MI8=
> ---- END SSH2 PUBLIC KEY ----
> ENDOFBUF
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

로컬 사용자 어카운트 삭제

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 로컬 사용자 어카운트를 삭제합니다.
Firepower-chassis /security # **delete local-user** *local-user-name*

단계 **3** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security #**commit-buffer**

다음 예에서는 foo 사용자 어카운트를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete local-user foo
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

로컬 사용자 어카운트 활성화 또는 비활성화

로컬 사용자 어카운트를 활성화하거나 비활성화하려면 사용자에게 관리자 또는 AAA 권한이 있어야 합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 활성화하거나 비활성화할 사용자의 로컬 사용자 보안 모드를 시작합니다.
Firepower-chassis /security # **scope local-user** *local-user-name*

단계 **3** 로컬 사용자 어카운트를 활성화할지 또는 비활성화할지 지정합니다.
Firepower-chassis /security/local-user # **set account-status** {**active** | **inactive**}

참고 관리자 사용자 어카운트는 항상 활성 상태로 설정됩니다. 수정할 수 없습니다.

다음 예에서는 어카운팅이라고 하는 로컬 사용자 어카운트를 활성화합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope local-user accounting
Firepower-chassis /security/local-user # set account-status active
```

로컬 인증 사용자에 대한 비밀번호 기록 지우기

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis # **scopesecurity**

단계 **2** 지정된 사용자 어카운트에 대한 로컬 사용자 보안 모드를 시작합니다.
Firepower-chassis /security # **scope local-user** *user-name*

단계 **3** 지정된 사용자 어카운트에 대한 비밀번호 기록을 지웁니다.
Firepower-chassis /security/local-user # **clear password-history**

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/local-user # **commit-buffer**

다음 예에서는 비밀번호 기록을 지우고 트랜잭션을 커밋합니다.

```
Firepower-chassis # scope security
Firepower-chassis /security # scope local-user admin
Firepower-chassis /security/local-user # clear password-history
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

6 장

이미지 관리

이미지 관리 정보

Firepower 4100/9300 섀시에서는 다음과 같은 두 가지 기본 이미지 유형을 사용합니다.

참고 모든 이미지는 보안 부팅을 통해 디지털로 서명되고 검증됩니다. 이미지를 수정하지 마십시오. 이미지를 수정하면 검증 오류를 수신하게 됩니다.

- 플랫폼 번들 — Firepower 플랫폼 번들은 Firepower Supervisor 및 Firepower 보안 모듈/엔진에서 작동하는 여러 개별 이미지가 모여 있는 컬렉션입니다. 플랫폼 번들은 Firepower eXtensible 운영 체제 소프트웨어 패키지입니다.
- 애플리케이션 — 애플리케이션 이미지는 Firepower 4100/9300 섀시의 보안 모듈/엔진에 구축할 소프트웨어 이미지입니다. 애플리케이션 이미지는 CSP(Cisco Secure Package) 파일로 전송되며, 논리적 디바이스를 생성하거나 이후 논리적 디바이스 생성에 대비하기 위해 보안 모듈/엔진에 구축될 때까지 수퍼바이저에 저장됩니다. Firepower Supervisor에 저장된 동일한 애플리케이션 이미지 유형에 대해 여러 가지 다른 버전이 있을 수 있습니다.

참고 플랫폼 번들 이미지와 하나 이상의 애플리케이션 이미지를 모두 업그레이드하려면 먼저 플랫폼 번들을 업그레이드해야 합니다.

Cisco.com에서 이미지 다운로드

시작하기 전에

Cisco.com 어카운트가 있어야 합니다.

절차

단계 1 웹 브라우저를 사용하여 http://www.cisco.com/go/firepower9300-software 또는 http://www.cisco.com/go/firepower4100-software로 이동합니다.
Firepower 4100/9300 섀시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.

단계 2 적절한 소프트웨어 이미지를 찾은 다음 로컬 컴퓨터에 다운로드합니다.

Firepower eXtensible 운영 체제 소프트웨어 이미지를 Firepower 4100/9300 섀시에 다운로드

FTP, SCP, SFTP 또는 TFTP를 사용하여 FXOS 소프트웨어 이미지를 Firepower 4100/9300 섀시에 복사할 수 있습니다.

시작하기 전에

컨피그레이션 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

- 이미지를 복사하고 있는 서버에 대한 IP 주소 및 인증 자격 증명
- FXOS 이미지 파일의 정규화된 이름

절차

단계 1 펌웨어 모드를 시작합니다.
Firepower-chassis # **scope firmware**

단계 2 FXOS 소프트웨어 이미지를 다운로드합니다.
Firepower-chassis /firmware # **download image** *URL*

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname/path/image_name*
- **scp**://*username@hostname/path/image_name*
- **sftp**://*username@hostname/path/image_name*
- **tftp**://*hostname*:*port-num*/*path*/*image_name*

단계 **3** 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.
Firepower-chassis /firmware # **show package** *image_name* **detail**

다음 예에서는 SCP 프로토콜을 사용하여 이미지를 복사합니다.

```
Firepower-chassis # scope firmware
Firepower-chassis /firmware # download image
scp://user@192.168.1.1/images/fxos-k9.1.1.1.119.SPA
Firepower-chassis /firmware # show package fxos-k9.1.1.1.119.SPA detail
Download task:
    File Name: fxos-k9.1.1.1.119.SPA
    Protocol: scp
    Server: 192.168.1.1
    Userid:
    Path:
    Downloaded Image Size (KB): 5120
    State: Downloading
    Current Task: downloading image fxos-k9.1.1.1.119.SPA from
192.168.1.1(FSM-STAGE:sam:dme:FirmwareDownloaderDownload:Local)
```

이미지 무결성 확인

이미지 무결성은 새로운 이미지가 Firepower 4100/9300 섀시에 추가되는 경우 자동으로 확인됩니다. 필요한 경우, 다음 절차를 사용하여 이미지 무결성을 수동으로 확인할 수 있습니다.

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 펌웨어 모드를 시작합니다.
Firepower-chassis# **scopefirmware**

단계 **3** 이미지를 나열합니다.
Firepower-chassis /firmware # **showpackage**

단계 **4** 이미지를 확인합니다.
Firepower-chassis /firmware # **verifyplatform-packversion** *version_number*

*version_number*는 확인 중인 FXOS 플랫폼 번들의 버전 번호입니다(예: 1.1(2.51)).

단계 **5** 확인하는 데 몇 분 정도 걸릴 수 있다는 메시지가 표시됩니다.
yes를 입력하여 확인을 계속할 것인지 확인합니다.

단계 **6** 다음 명령을 사용하여 이미지 확인 상태를 점검합니다.
Firepower-chassis /firmware # **showvalidate-task**

Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드

시작하기 전에

Cisco.com에서 플랫폼 번들 소프트웨어 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 54 페이지 참조) 한 다음 해당 이미지를 Firepower 4100/9300 섀시에 다운로드합니다(Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 57 페이지 참조).

참고 업그레이드 프로세스는 일반적으로 20~30분이 소요됩니다.

독립형 논리적 디바이스를 실행 중인 Firepower 9300 또는 Firepower 4100 Series 보안 어플라이언스를 업그레이드하는 중인 경우 또는 섀시 내 클러스터를 실행 중인 Firepower 9300 보안 어플라이언스를 업그레이드하는 중인 경우, 트래픽은 업그레이드 중에 디바이스를 통과하지 않습니다.

섀시 간 클러스터의 일부인 Firepower 9300 또는 Firepower 4100 Series 보안 어플라이언스를 업그레이드하는 중인 경우, 트래픽은 업그레이드 중에 업그레이드되고 있는 디바이스를 통과하지 않습니다. 그러나 클러스터에 있는 다른 디바이스는 트래픽을 계속 전달합니다.

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 펌웨어 모드를 시작합니다.
Firepower-chassis# **scopefirmware**

단계 **3** 자동 설치 모드를 시작합니다.
Firepower-chassis /firmware # **scopeauto-install**

단계 **4** FXOS 플랫폼 번들을 설치합니다.
Firepower-chassis /firmware/auto-install # **installplatformplatform-vers** *version_number*

*version_number*는 설치 중인 FXOS 플랫폼 번들의 버전 번호입니다(예: 1.1(2.51)).

단계 **5** 시스템은 설치할 소프트웨어 패키지를 먼저 확인합니다. 시스템은 현재 설치된 애플리케이션과 지정된 FXOS 플랫폼 소프트웨어 패키지가 호환되지 않는지 여부를 알려줍니다. 또한 기존 세션이 종료되고 시스템이 업그레이드의 일부로 재부팅되어야 한다고 경고합니다.
yes를 입력하여 확인을 계속할 것인지 확인합니다.

단계 **6** **yes**를 입력하여 설치를 계속할 것인지 확인하거나 **no**를 입력하여 설치를 취소합니다.
Firepower eXtensible 운영 체제에서는 번들의 압축을 풀고 구성 요소를 업그레이드하거나 다시 로드합니다.

단계 **7** 업그레이드 프로세스를 모니터링하려면 다음을 수행합니다.

a) **scopefirmware**를 입력합니다.

b) **scopeauto-install**를 입력합니다.
c) **showfsmstatusexpand**를 입력합니다.

Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드

FTP, SCP, SFTP 또는 TFTP를 사용하여 논리적 디바이스 소프트웨어 이미지를 Firepower 4100/9300 섀시에 복사할 수 있습니다.

시작하기 전에

컨피그레이션 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

- 이미지를 복사하고 있는 서버에 대한 IP 주소 및 인증 자격 증명
- 소프트웨어 이미지 파일의 정규화된 이름

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower-chassis # **scopessa**

단계 **2** 애플리케이션 소프트웨어 모드를 시작합니다.
Firepower-chassis /ssa # **scopeapp-software**

단계 **3** 논리적 디바이스 소프트웨어 이미지를 다운로드합니다.
Firepower-chassis /ssa/app-software # **download image** *URL*

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*
- **scp**://*username@hostname*/*path*
- **sftp**://*username@hostname*/*path*
- **tftp**://*hostname*:*port-num*/*path*

단계 **4** 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.
Firepower-chassis /ssa/app-software # **show download-task**

단계 **5** 다음 명령을 사용하여 다운로드한 애플리케이션을 확인합니다.
Firepower-chassis /ssa/app-software # **up**
Firepower-chassis /ssa # **show app**

단계 **6** 다음 명령을 사용하여 특정 애플리케이션에 대한 세부사항을 확인합니다.
Firepower-chassis /ssa # **scope app** *application_type image_version*

Firepower-chassis /ssa/app # **show expand**

다음 예에서는 SCP 프로토콜을 사용하여 이미지를 복사합니다.

```
Firepower-chassis # scope ssa
Firepower-chassis /ssa # scope app-software
Firepower-chassis /ssa/app-software # download image
scp://user@192.168.1.1/images/cisco-asa.9.4.1.65.csp
Firepower-chassis /ssa/app-software # show download-task

Downloads for Application Software:
    File Name                     Protocol   Server               Userid          State
    ----------------------------- ---------- -------------------- --------------- -----
    cisco-asa.9.4.1.65.csp        Scp        192.168.1.1          user            Downloaded

Firepower-chassis /ssa/app-software # up

Firepower-chassis /ssa # show app

Application:
    Name       Version    Description Author     Deploy Type CSP Type    Is Default App
    ---------- ---------- ----------- ---------- ----------- ----------- --------------
    asa        9.4.1.41   N/A                    Native      Application No
    asa        9.4.1.65   N/A                    Native      Application Yes

Firepower-chassis /ssa # scope app asa 9.4.1.65
Firepower-chassis /ssa/app # show expand

Application:
    Name: asa
    Version: 9.4.1.65
    Description: N/A
    Author:
    Deploy Type: Native
    CSP Type: Application
    Is Default App: Yes

    App Attribute Key for the Application:
        App Attribute Key Description
        ----------------- -----------
        cluster-role      This is the role of the blade in the cluster
        mgmt-ip           This is the IP for the management interface
        mgmt-url          This is the management URL for this application

    Net Mgmt Bootstrap Key for the Application:
        Bootstrap Key Key Data Type Is the Key Secret Description
        ------------- ------------- ----------------- -----------
        PASSWORD      String        Yes               The admin user password.

    Port Requirement for the Application:
        Port Type: Data
        Max Ports: 120
        Min Ports: 1

        Port Type: Mgmt
        Max Ports: 1
        Min Ports: 1

        Mgmt Port Sub Type for the Application:
            Management Sub Type
            -------------------
            Default

        Port Type: Cluster
        Max Ports: 1
        Min Ports: 0
Firepower-chassis /ssa/app #
```

논리적 디바이스를 위한 이미지 버전 업데이트

시작하기 전에

참고 Firepower Threat Defense 논리적 디바이스의 초기 생성 이후에 Firepower Threat Defense 논리적 디바이스를 Firepower Chassis Manager 또는 FXOS CLI를 사용하여 업그레이드하지 마십시오. Firepower Threat Defense 논리적 디바이스를 업그레이드하려면 Firepower Management Center를 사용해야 합니다. 자세한 내용은 Firepower System 릴리스 노트를 참조하십시오. http://www.cisco.com/c/en/us/support/security/defense-center/products-release-notes-list.html

또한, Firepower Threat Defense 논리적 디바이스에 대한 업데이트는 **Logical Devices(**논리적 디바이스**)** > **Edit(**편집**)** 및 **System(**시스템**)** > **Updates(**업데이트**)** 페이지(Firepower Chassis Manager의 페이지)에 반영되지 않습니다. 이러한 페이지에서 표시된 버전은 Firepower Threat Defense 논리적 디바이스를 생성하는 데 사용된 소프트웨어 버전(CSP 이미지)을 나타냅니다.

Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 54 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 다운로드합니다(Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 57 페이지 참조).

플랫폼 번들 이미지와 하나 이상의 애플리케이션 이미지를 모두 업그레이드하려면 먼저 플랫폼 번들을 업그레이드해야 합니다.

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower-chassis # **scopessa**

단계 **2** 업데이트 중인 보안 모듈의 범위를 설정합니다.
Firepower-chassis /ssa # **scopeslot** *slot_number*

단계 **3** 업데이트 중인 애플리케이션의 범위를 설정합니다.
Firepower-chassis /ssa/slot # **scopeapp-instance** *app_template*

단계 **4** 시작 버전을 업데이트할 버전으로 설정합니다.
Firepower-chassis /ssa/slot/app-instance # **setstartup-version** *version_number*

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다. 애플리케이션 이미지가 업데이트되고 애플리케이션이 다시 시작됩니다.

다음 예에서는 보안 모듈 1에서 실행 중인 ASA의 소프트웨어 이미지를 업데이트합니다. show 명령을 사용하여 업데이트 상태를 확인할 수 있습니다.

```
Firepower-chassis# scope ssa
Firepower-chassis /ssa # scope slot 1
Firepower-chassis /ssa/slot # scope app-instance asa
Firepower-chassis /ssa/slot/app-instance # set startup-version 9.4.1.65
Firepower-chassis /ssa/slot/app-instance* # show configuration pending
 enter app-instance asa
+    set startup-version 9.4.1.65
 exit
Firepower-chassis /ssa/slot/app-instance* # commit-buffer
Firepower-chassis /ssa/slot/app-instance # show

Application Instance:
    Application Name Admin State Operational State Running Version Startup Version
    ---------------- ----------- ----------------- --------------- ---------------
    asa              Enabled     Updating          9.4.1.41        9.4.1.65
Firepower-chassis /ssa/slot/app-instance #
Firepower-chassis /ssa/slot/app-instance # show

Application Instance:
    Application Name Admin State Operational State Running Version Startup Version
    ---------------- ----------- ----------------- --------------- ---------------
    asa              Enabled     Online            9.4.1.65        9.4.1.65
Firepower-chassis /ssa/slot/app-instance #
```

펌웨어 업그레이드

다음 절차를 사용하여 Firepower 4100/9300 섀시에서 펌웨어를 업그레이드합니다.

절차

단계 **1** 웹 브라우저를 사용하여 http://www.cisco.com/go/firepower9300-software 또는 http://www.cisco.com/go/firepower4100-software로 이동합니다.
Firepower 4100/9300 섀시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.

단계 **2** Cisco.com에서 적절한 펌웨어 패키지를 찾은 다음 Firepower 4100/9300 섀시에서 액세스할 수 있는 서버로 다운로드합니다.

단계 **3** Firepower 4100/9300 섀시에서 펌웨어 모드를 시작합니다.
Firepower-chassis # **scopefirmware**

단계 **4** FXOS 펌웨어 이미지를 Firepower 4100/9300 섀시에 다운로드합니다.
Firepower-chassis /firmware # **download image** *URL*

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**:// *username@hostname* / *path*
- **scp**:// *username@hostname* / *path*
- **sftp**:// *username@hostname* / *path*
- **tftp**:// *hostname* : *port-num* / *path*

단계 **5** 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.

Firepower-chassis /firmware # **show download task** *image_name* **detail**

단계 **6** 다운로드가 완료되면 다음 명령을 입력하여 펌웨어 패키지의 콘텐츠를 볼 수 있습니다.
Firepower-chassis /firmware # **show package** *image_name* **expand**

단계 **7** 다음 명령을 입력하여 펌웨어 패키지의 버전 번호를 볼 수 있습니다.
Firepower-chassis /firmware # **show package**

이 버전 번호는 펌웨어 패키지를 설치할 때 다음 단계에서 사용됩니다.

단계 **8** 펌웨어 패키지를 설치합니다.

a) 펌웨어 설치 모드를 시작합니다.
Firepower-chassis /firmware # **scope firmware-install**

b) 펌웨어 패키지를 설치합니다.
Firepower-chassis /firmware/firmware-install # **install firmware pack-version** *version_number*

시스템에서는 펌웨어 패키지를 검증하며, 검증 프로세스를 완료하는 데 몇 분 정도 걸릴 수 있다고 알립니다.

c) 검증을 계속하려면 **yes**를 입력합니다.
시스템에서는 펌웨어 패키지를 검증한 후, 설치 프로세스를 완료하는 데 몇 분 정도 걸릴 수 있으며 업데이트 프로세스 도중 시스템이 재부팅될 것이라고 알립니다.

d) 설치를 계속하려면 **yes**를 입력합니다. 업그레이드 프로세스 도중 Firepower 4100/9300 섀시의 전원을 껐다가 다시 켜지 마십시오.

단계 **9** 업그레이드 프로세스를 모니터링하려면 다음을 수행합니다.
Firepower-chassis /firmware/firmware-install # **show detail**

단계 **10** 설치가 완료되고 나면 다음 명령을 입력하여 현재 펌웨어 버전을 볼 수 있습니다.
Firepower-chassis /firmware/firmware-install # **top**

Firepower-chassis # **scope chassis 1**

Firepower-chassis /firmware # **show sup version**

다음 예에서는 펌웨어를 버전 1.0.10으로 업그레이드합니다.

```
Firepower-chassis# scope firmware
Firepower-chassis /firmware # download image
tftp://10.10.10.1/fxos-k9-fpr9k-firmware.1.0.10.SPA
Firepower-chassis /firmware # show download-task fxos-k9-fpr9k-firmware.1.0.10.SPA detail

Download task:
    File Name: fxos-k9-fpr9k-firmware.1.0.10.SPA
    Protocol: Tftp
    Server: 10.10.10.1
    Port: 0
    Userid:
    Path:
    Downloaded Image Size (KB): 2104
    Time stamp: 2015-12-04T23:51:57.846
    State: Downloading
    Transfer Rate (KB/s): 263.000000
    Current Task: unpacking image fxos-k9-fpr9k-firmware.1.0.10.SPA on primary(
FSM-STAGE:sam:dme:FirmwareDownloaderDownload:UnpackLocal)

Firepower-chassis /firmware # show package fxos-k9-fpr9k-firmware.1.0.10.SPA expand
```

```
Package fxos-k9-fpr9k-firmware.1.0.10.SPA:
    Images:
        fxos-k9-fpr9k-fpga.1.0.5.bin
        fxos-k9-fpr9k-rommon.1.0.10.bin

Firepower-chassis /firmware # show package

Name                                       Version
------------------------------------------ -------
fxos-k9-fpr9k-firmware.1.0.10.SPA          1.0.10

Firepower-chassis /firmware # scope firmware-install
Firepower-chassis /firmware/firmware-install # install firmware pack-version 1.0.10

Verifying FXOS firmware package 1.0.10. Verification could take several minutes.
Do you want to proceed? (yes/no):yes

FXOS SUP ROMMON: Upgrade from 1.0.10 to 1.0.10
FXOS SUP FPGA  : Upgrade from 1.04 to 1.05

This operation upgrades SUP firmware on Security Platform.
Here is the checklist of things that are recommended before starting the install operation
(1) Review current critical/major faults
(2) Initiate a configuration backup

Attention:
   The system will be reboot to upgrade the SUP firmware.
   The upgrade operation will take several minutes to complete.
   PLEASE DO NOT POWER RECYCLE DURING THE UPGRADE.
Do you want to proceed? (yes/no):yes

Upgrading FXOS SUP firmware software package version 1.0.10

command executed
```

7 장

보안 인증 컴플라이언스

보안 인증 컴플라이언스

미국 연방 정부 기관에서는 경우에 따라 미국 국방부 및 글로벌 인증 기관에서 수립한 보안 표준을 준수하는 장비와 소프트웨어만 사용해야 합니다. Firepower 4100/9300 섀시는 이러한 여러 보안 인증 표준에 대해 컴플라이언스를 지원합니다.

이러한 표준에 대한 컴플라이언스를 지원하는 기능을 활성화하는 단계에 대해서는 다음 항목을 참조하십시오.

- FIPS 모드 활성화, 64 페이지
- Common Criteria 모드 활성화, 65 페이지
- IPSec 보안 채널 구성, 66 페이지

- 트러스트 포인트에 대한 정적 CRL 구성, 71 페이지
- 인증서 해지 목록 확인 정보, 72 페이지
- CRL의 주기적인 다운로드 구성, 77 페이지
- NTP 서버 인증 활성화, 79 페이지
- LDAP 키 링 인증서 설정, 79 페이지
- IP 액세스 목록 구성, 80 페이지
- 클라이언트 인증서 인증 활성화, 81 페이지
- 최소 비밀번호 길이 확인 구성, 45 페이지
- 최대 로그인 시도 횟수 설정, 43 페이지
- 사용자 역할, 37 페이지

참고 이러한 항목에서는 Firepower 4100/9300 섀시에서의 인증 컴플라이언스 활성화에 대해서만 설명합니다. Firepower 4100/9300 섀시에서 인증 컴플라이언스를 활성화하더라도 연결된 논리적 디바이스에 컴플라이언스가 자동으로 전파되지는 않습니다.

FIPS 모드 활성화

Firepower 4100/9300 섀시에서 FIPS 모드를 활성화하려면 다음 단계를 수행합니다.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesystem

scopesecurity

단계 **2** FIPS 모드를 활성화합니다.
enablefips-mode

단계 **3** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **4** 시스템을 재부팅합니다.
connectlocal-mgmt

reboot

다음에 할 작업

FXOS 릴리스 2.0.1 이전 버전의 경우 디바이스의 첫 번째 설정 시 생성된 기존 SSH 호스트 키가 1024 비트로 하드 코딩되어 있습니다. FIPS 및 Common Criteria 인증 요건을 준수하려면 기존의 호스트 키를 삭제하고 SSH 호스트 키 생성, 65 페이지에 자세히 나와 있는 절차를 사용하여 새로운 호스트 키를 생성해야 합니다. 이러한 추가 단계를 수행하지 않으면 FIPS 모드가 활성화된 상태에서 디바이스가 재부팅된 이후에 SSH를 사용하여 수퍼바이저에 연결할 수 없습니다. FXOS 2.0.1 이상을 사용하여 초기 설정을 수행한 경우 새 호스트 키를 생성할 필요가 없습니다.

Common Criteria 모드 활성화

Firepower 4100/9300 섀시에서 Common Criteria 모드를 활성화하려면 다음 단계를 수행하십시오.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesystem

scopesecurity

단계 **2** Common Criteria 모드를 활성화합니다.
enablecc-mode

단계 **3** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **4** 시스템을 재부팅합니다.
connectlocal-mgmt

reboot

다음에 할 작업

FXOS 릴리스 2.0.1 이전 버전의 경우 디바이스의 첫 번째 설정 시 생성된 기존 SSH 호스트 키가 1024 비트로 하드 코딩되어 있습니다. FIPS 및 Common Criteria 인증 요건을 준수하려면 기존의 호스트 키를 삭제하고 **SSH** 호스트 키 생성**, 65** 페이지에 자세히 나와 있는 절차를 사용하여 새로운 호스트 키를 생성해야 합니다. 이러한 추가 단계를 수행하지 않으면 Common Criteria 모드가 활성화된 상태에서 디바이스가 재부팅된 이후에 SSH를 사용하여 수퍼바이저에 연결할 수 없습니다. FXOS 2.0.1 이후 버전을 사용하여 초기 설정을 수행한 경우, 호스트 키를 새로 생성할 필요가 없습니다.

SSH 호스트 키 생성

FXOS 릴리스 2.0.1 이전 버전의 경우 디바이스의 초기 설정 시 생성된 기존 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 공통 Common Criteria을 준수하려면 기존의 호스트 키를 삭제하

고 새로운 호스트 키를 생성해야 합니다. 자세한 내용은 FIPS 모드 활성화, 64 페이지 또는 Common Criteria 모드 활성화, 65 페이지를 참고하십시오.

기존의 SSH 호스트 키를 삭제하고 새로운 인증-준수 호스트 키를 생성하려면 다음 단계를 수행합니다.

절차

단계 **1** FXOS CLI에서 서비스 모드를 시작합니다.
scopesystem

scopeservices

단계 **2** SSH 호스트 키를 삭제합니다.
deletessh-serverhost-key

단계 **3** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **4** SSH 호스트 키 크기를 2048비트로 설정합니다.
setssh-serverhost-keyrsa 2048

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **6** 새로운 SSH 호스트 키를 생성합니다.
createssh-serverhost-key

commit-buffer

단계 **7** 새로운 호스트 키 크기를 확인합니다.
showssh-serverhost-key

호스트 키 크기: 2048

IPSec 보안 채널 구성

공용 네트워크를 통과하는 데이터 패킷에 대해 엔드 투 엔드 암호화 및 인증 서비스를 제공하기 위해 Firepower 4100/9300 섀시에서 IPSec를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증 컴플라이언스, 63 페이지를 참조하십시오.

참고 IKE와 SA 연결 간에 일치하는 암호화 키 보안 강도가 적용되도록 구성하는 경우(아래 절차에서 sa-strength-enforcement를 yes(예)로 설정):

SA 적용이 활성화된 경우	IKE 협상 키 크기가 ESP 협상 키 크기보다 작은 경우, 연결이 실패합니다. IKE 협상 키 크기가 ESP 협상 키 크기 이상인 경우, SA 적용 확인을 통과하며 연결이 성공합니다.
SA 적용이 비활성화된 경우	SA 적용 확인을 통과하며 연결이 성공합니다.

IPSec 보안 채널을 구성하려면 다음 단계를 수행합니다.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesystem

scopesecurity

단계 **2** 키 링을 생성합니다.
enterkeyringssp

!createcertreqsubject-name *subject-name***ip** *ip*

단계 **3** 연결된 인증서 요청 정보를 입력합니다.
entercertreq

단계 **4** 국가를 설정합니다.
setcountry *country*

단계 **5** DNS를 설정합니다.
setdns *dns*

단계 **6** 이메일을 설정합니다.
sete-mail *email*

단계 **7** IP 정보를 설정합니다.
setfi-a-ip *fi-a-ip*

setfi-a-ipv6 *fi-a-ipv6*

setfi-b-ip *fi-b-ip*

setfi-b-ipv6 *fi-b-ipv6*

setipv6 *ipv6*

단계 **8** 지역을 설정합니다.

setlocality *locality*

단계 **9** 조직 이름을 설정합니다.
setorg-name *org-name*

단계 **10** 조직 단위 이름을 설정합니다.
setorg-unit-name *org-unit-name*

단계 **11** 비밀번호를 설정합니다.
!setpassword

단계 **12** 상태를 설정합니다.
setstate *state*

단계 **13** certreq에 대한 주체 이름을 설정합니다.
setsubject-name *subject-name*

단계 **14** 종료합니다.
exit

단계 **15** 모듈러스를 설정합니다.
setmodulus *modulus*

단계 **16** 인증서 요청에 대한 재생성을 설정합니다.
setregenerate *{ yes | no }*

단계 **17** 트러스트 포인트를 설정합니다.
settrustpointinterca

단계 **18** 종료합니다.
exit

단계 **19** 새로 생성한 트러스트 포인트를 입력합니다.
entertrustpointinterca

단계 **20** 인증서 서명 요청을 생성합니다.
setcertchain

예제**:**

```
-----BEGIN CERTIFICATE-----
MIIF3TCCA8WgAwIBAgIBADANBgkqhkiG9w0BAQsFADBwMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExDDAKBgNVBAcMA1NKQzEOMAwGA1UECgwFQ2lzY28xDTALBgNV
BAsMBFNUQlUxCzAJBgNVBAMMAkNBMRowGAYJKoZIhvcNAQkBFgtzc3BAc3NwLm5l
dDAeFw0xNjEyMDgxOTMzNTJaFw0yNjEyMDYxOTMzNTJaMHAxCzAJBgNVBAYTAlVT
MQswCQYDVQQIDAJDQTEMMAoGA1UEBwwDU0pDMQ4wDAYDVQQKDAVDaXNjbzENMAsG
A1UECwwEU1RCVTELMAkGA1UEAwwCQ0ExGjAYBgkqhkiG9w0BCQEWC3NzcEBzc3Au
bmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA2ukWyMLQuLqTvhq7
zFb3Oz/iyDG/ui6mrLIYn8wE3E39XcXA1/x9IHCmxFKNJdD7EbsggfOuy0Bj+Y4s
+uZ1VapBXV/JrAie7bNn3ZYrI29yuyOrIqoi9k9gL/oRBzH18BwBwGHBOz3hGrSK
Yc2yhsq9y/6yI3nSuLZm6ybmUKjTa+B4YuhDTz4hl/I9x/J5nbGiab3vLDKss1nO
xP9+1+Lc690Vl8/mNPWdjCjDI+U/L9keYs/rbZdRSeXy9kMae42+4FlRHDJjPcSN
Yw1g/gcR2F7QUKRygKckJKXDX2QIiGYSctlSHj18O87o5s/pmQAWWRGkKpfDv3oH
cMPgI2T9rC0D8NNcgPXj9PFKfexoGNGwNTO85fK3kjgMOdWbdeMG3EihxEEOUPD0
```

```
Fdu0HrTM5lvwb+vr5wE9HsAiMJ8UuujmHqH5mlwyy3Me+cEDHo0hLeNs+AFrqEXQ
e9S+KZC/dq/9zOLpRsVqSfJsAuVl/QdPDbWShjflE/fP2Wj01PqXywQydzymVvgE
wEZaoFg+mlGJm0+q4RDvnpzEviOYNSAGmOkILh5HQ/eYDcxvd0qbORWb31H32ySl
Ila6UTT9+vnND1f838fxvNvr8nyGD2S/LVaxnZlO4jcSIvtdizbbT8u5B4VcLKIC
x0vkqjo6RvNZJ52sUaD9C3UodTUCAwEAAaOBgTB/MC8GA1UdHwQoMCYwJKAioCCG
Hmh0dHA6Ly8xOTIuMTY4LjQuMjkvcm9vdGNhLmNybDAdBgNVHQ4EFgQU7Jg01A74
jpx8U0APk76pVfYQQ5AwHwYDVR0jBBgwFoAU7Jg01A74jpx8U0APk76pVfYQQ5Aw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAvI8ky2jiXc4wPiMuxIfY
W7DRmszPUWQ7edor7yxuCqzHLVFFOwYRudsyXbv7INR3rJ/X1cRQj9+KidWWVxpo
pFahRhzYxVZ10DHKIzGTQS3jiHgrF3Z8ohWbL15L7PEDlrxMBoJvabPeQRgTmY/n
XZJ7qRYbypO3gUMCaCZ12raJc3/DIpBQ29yweCbUkc9qiHKA0IbnvAxoroHWmBld
94LrJCggfMQTuNJQszJiVVsYJfZ+utlDp2QwfdDv7B0JkwTBjdwRSfotEbc5R18n
BNXYHqxuoNMmqbS3KjCLXcH6xIN8t+UkfP89hvJt/fluJ+s/VJSVZWK4tAWvR7wl
QngCKRJW6FYpzeyNBctiJ07wO+Wt4e3KhIjJDYvA9hFixWcVGDf2r6QW5BYbgGOK
DkHb/gdr/bcdLBKN/PtSJ+prSrpBSaA6rJX8D9UmfhqqN/3f+sS1fM4qWORJc6G2
gAcg7AjEQ/0do512vAI8p8idOg/Wv1O17mavzLpcue05cwMCX9fKxKZZ/+7Pk19Y
ZrXS6uMn/CGnViptn0w+uJ1IRj1oulk+/ZyPtBvFHUkFRnhoWj5SMFyds2IaatyI
47N2ViaZBxhU3GICaH+3O+8rs9Kkz9tBZDSnEJVZA6yxaNCVP1bRUO20G3oRTmSx
8iLBjN+BXggxMmG8ssHisgw=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFqDCCA5CgAwIBAgIBBDANBgkqhkiG9w0BAQsFADBwMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExDDAKBgNVBAcMA1NKQzEOMAwGA1UECgwFQ2lzY28xDTALBgNV
BAsMBFNUQlUxCzAJBgNVBAMMAkNBMRowGAYJKoZIhvcNAQkBFgtzc3BAc3NwLm5l
dDAeFw0xNjEyMTUyMTM0NTRaFw0yNjEyMTMyMTM0NTRaMHwxCzAJBgNVBAYTAlVT
MQswCQYDVQQIDAJDQTEPMA0GA1UECgwGbmV3c3RnMRAwDgYDVQQLDAduZXdzdGJ1
MRMwEQYDVQQDDApbnRlcm0xLWNhMSgwJgYJKoZIhvcNAQkBFhlpbnRlcm0xLWNh
QGludGVybTEtY2EubmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA
wLpNnyEx5I4P8uDoWKWF3IZsegjhLANsodxuAUmhmwKekd0OpZZxHMw1wSO4IBX5
4itJS0xyXFzPmeptG3OXvNqCcsT+4BXl3DoGgPMULccc4NesHeg2z8+q3SPA6uZh
iseWNvKfnUjixbQEBtcrWBiSKnZuOz1cpuBn34gtgeFFoCEXN+EZVpPESiancDVh
8pCPlipc/08ZJ3o9GW2j0eHJN84sguIEDL812ROejQvpmfqGUq11stkIIuh+wB+V
VRhUBVG7pV57I6DHeeRp6cDMLXaM3iMTelhdShyo5YUaRJMak/t8kCqhtGXfuLlI
E2AkxKXeeveR9n6cpQd5JiNzCT/t9IQL/T/CCqMICRXLFpLCS9o5S5O2B6QFgcTZ
yKR6hsmwe22wpK8QI7/5oWNXlolb96hHJ7RPbG7RXYqmcLiXY/d2j9/RuNoPJawI
hLkfhoIdPA28xlnfIB1azCmMmdPcBO6cbUQfCj5hSmk3StVQKgJCjaujz55TGGd1
GjnxDMX9twwz7Ee51895Xmtr24qqaCXJoW/dPhcIIXRdJPMsTJ4yPG0BieuRwd0p
i8w/rFwbHzv4C9Fthw1JrRxH1yeHJHrLlZgJ5txSaVUIgrgVCJaf6/jrRRWoRJwt
AzvnzYql2dZPCcEAYgP7JcaQpvdpuDgq++NgBtygiqECAwEAAaNBMD8wDAYDVR0T
BAUwAwEB/zAvBgNVHR8EKDAmMCSgIqAghh5odHRwOi8vMTkyLjE2OC40LjI5L2lu
dGVybS5jcmwwDQYJKoZIhvcNAQELBQADggIBAG/XujJh5G5UWo+cwTSitAezWbJA
h1dAiXZ/OYWZSxkFRliErKdupLqL0ThjnX/wRFfEXbrBQwm5kWAUUDr97D1Uz+2A
8LC5I8SWKXmyf0jUtsnEQbDZb33oVL7yXJk/A0SF0jihpPheMA+YRazalT9xj9KH
PE7nHCJMbb2ptrHUyvBrKSYrSeEqOpQU2+otnFyV3rS9aelgVjuaWyaWOc3lZ1Oi
CC2tJvY3NnM56j5iesxUCeY/SZ2/ECXN7RRBViLHmA3gFKmWf3xeNiKkxmJCxOaa
UWPC1x2V66I8DG9uUzlWyd79O2dy52aAphAHC6hqlzb6v+gw1Tld7UxaqVd8CD5W
ATjNs+ifkJS1h5ERxHjgcurZXOpR+NWpwF+UDzbMXxx+KAAXCI6ltCd8Pb3wOUC3
PKvwEXaIcCcxGx71eRLpWPZFyEoi4N2NGE9OXRjz0K/KERZgNhsIW3bQMjcw3aX6
OXskEuKgsayctnWyxVqNnqvpuz06kqyubh4+ZgGKZ5LNEXYmGNz3oED1rUN636Tw
SjGAPHgeROzyTFDixCei6aROlGdP/Hwvb0/+uThIe89g8WZ0djTKFUM8uBO3f+II
/cbuyBO1+JrDMq8NkAjxKlJlp1c3WbfCue/qcwtcfUBYZ4i53a56UNF5Ef0rpy/8
B/+07Me/p2y9Luqa
-----END CERTIFICATE-----
ENDOFBUF
```

단계 **21** 인증서 서명 요청을 표시합니다.

showcertreq

예제:

```
Firepower-chassis# /security/keyring # show certreq
Certificate request subject name: SSP
Certificate request ip address: 192.168.0.111
Certificate request FI A ip address: 0.0.0.0
Certificate request FI B ip address: 0.0.0.0
Certificate request e-mail name:
Certificate request ipv6 address: ::
Certificate request FI A ipv6 address: ::
Certificate request FI B ipv6 address: ::
Certificate request country name: US
State, province or county (full name): CA
Locality name (eg, city): SJC
Organisation name (eg, company): Cisco
Organisational Unit Name (eg, section): Sec
DNS name (subject alternative name):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIICwTCCAakCAQAwVTELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAkNBMQwwCgYDVQQH
DANTSkMxDjAMBgNVBAoMBUNpc2NvMQ0wCwYDVQQLDARTVEJVMQwwCgYDVQQDDANT
U1AwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDq292Rq3t0laoxPbfE
p/lTKr6rxFhPqSSbtm6sXer//VZFiDTWODockDItuf4Kja215mIS0RyvEYVeRgAs
wbN459wm0BASd8xCjIhsuHDV7yHu539BnvRW6Q2o+gHeSRwckqjClK/tsIxsPkV0
6OduZYXk2bnsLWs6tNk3uzOIT2Q0FcZ1ET66C8fyyKWTrmvcZjDjkMm2nDFsPlX9
39TYPItDkJE3PocqyaCqmT4uobOuvQeLJh/efkBvwhb4BF8vwzRpHWTdjjU5YnR1
qiR4q7j1RmzVFxCDY3IVP/KDBoa5NyCLEUZECP5QCQFDzIRETZwVOKtxUVG0Nljd
K5TxAgMBAAGgJzAlBgkqhkiG9w0BCQ4xGDAWMBQGA1UdEQQNMAuCA1NTUIcEwKgA
rjANBgkqhkiG9w0BAQsFAAOCAQEARtRBoInxXkBYNlVeEoFCqKttu3+Hc7UdyoRM
2L2pjx5OHbQICC+8NRVRMYujTnp67BWuUZZl03dGP4/lbN6bC9P3CvkZdKUsJkN0
m1Ye9dgz7MO/KEcosarmoMl9WB8LlweVdt6ycSdJzs9shOxwT6TAZPwL7gq/1ShF
RJh6sq5W9p6E0SjYefK62E7MatRjDjS8DXoxj6gfn9DqK15iVpkK2QqT5rneSGj+
R+20TcUnT0h/S5K/bySEM/3U1gFxQCOzbzPuHkj28kXAVczmTxXEkJBFLVduWNo6
DT3u0xImiPR1sqW1jpMwbhC+ZGDtvgKjKHToagup9+8R9IMcBQ==
-----END CERTIFICATE REQUEST-----
```

단계 **22** IPSec 모드로 들어갑니다.
scopeipsec

단계 **23** 로그 상세 레벨을 설정합니다.
setlog-level *log_level*

단계 **24** IPSec 연결을 생성하고 입력합니다.
enterconnection *connection_name*

단계 **25** IPSec 모드를 터널 또는 전송으로 설정합니다.
setmode *tunnel_or_transport*

단계 **26** 로컬 IP 주소를 설정합니다.
setlocal-addr *ip_address*

단계 **27** 원격 IP 주소를 설정합니다.
setremote-addr *ip_address*

단계 **28** 터널 모드를 사용하는 경우, 원격 서브넷을 설정합니다.

setremote-subnet *ip/mask*

단계 **29** (선택 사항) 원격 ID를 설정합니다.
setremote-ike-ident *remote_identity_name*

단계 **30** 키 링 이름을 설정합니다.
setkeyring-name *name*

단계 **31** (선택 사항) 키 링 비밀번호를 설정합니다.
setkeyring-passwd *passphrase*

단계 **32** (선택 사항) IKE-SA 수명을 분 단위로 설정합니다.
setike-rekey-time *minutes*

minutes 값은 60~1440의 정수일 수 있습니다.

단계 **33** (선택 사항) 하위 SA 수명을 분 단위(30~480분)로 설정합니다.
setesp-rekey-time *minutes*

minutes 값은 30~480의 정수일 수 있습니다.

단계 **34** (선택 사항) 초기 연결 시 수행할 재전송 시퀀스 수를 설정합니다.
setkeyringtries *retry_number*

retry_number 값은 1~5의 정수일 수 있습니다.

단계 **35** (선택 사항) 인증서 해지 목록 확인을 활성화하거나 비활성화합니다.
setrevoke-policy *{ relaxed | strict }*

단계 **36** 연결을 활성화합니다.
setadmin-stateenable

단계 **37** 모든 연결을 다시 로드합니다.
reload-conns

단계 **38** (선택 사항) IPsec에 기존의 트러스트 포인트 이름을 추가합니다.
createauthority *trustpoint_name*

단계 **39** IKE와 SA 연결 간에 일치하는 암호화 키 보안 강도가 적용되도록 구성합니다.
setsa-strength-enforcement *yes_or_no*

트러스트 포인트에 대한 정적 CRL 구성

해지된 인증서는 CRL(Certification Revocation List)에 유지됩니다. 클라이언트 애플리케이션은 CRL을 사용하여 서버의 인증을 확인합니다. 서버 애플리케이션은 CRL을 활용하여 더 이상 신뢰할 수 없는 클라이언트 애플리케이션의 액세스 요청을 수락 또는 거부합니다.

CRL(Certification Revocation List) 정보를 사용하여 피어 인증서를 검증하도록 Firepower 4100/9300 섀시를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증 컴플라이언스, 63 페이지를 참조하십시오.

CRL 정보를 사용하여 피어 인증서를 검증하려면 다음 단계를 수행합니다.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesecurity

단계 **2** 트러스트 포인트 모드를 시작합니다.
scopetrustpoint *trustname*

단계 **3** 취소 모드를 시작합니다.
scoperevoke

단계 **4** CRL 파일을 다운로드합니다.
importcrl *protocol*://*user_id*@*CA_or_CRL_issuer_IP*/tmp/*DoDCA1CRL1.crl*

단계 **5** (선택 사항) CRL 정보 가져오기 프로세스의 상태를 표시합니다.
showimport-taskdetail

단계 **6** 인증서 해지 방법을 CRL 전용으로 설정합니다.
setcertrevokemethod{crl}

인증서 해지 목록 확인 정보

CRL(Certificate Revocation List, 인증서 해지 목록) 확인 모드를 IPSec, HTTPS 및 보안 LDAP 연결에서 Strict(엄격) 또는 Relaxed(완화) 중 하나로 구성할 수 있습니다.

동적(비정적) CRL 정보는 X.509 인증서의 CDP 정보에서 수집되며 동적 CRL 정보를 나타냅니다. 정적 CRL 정보는 시스템 관리를 통해 수동으로 다운로드되며 FXOS 시스템의 로컬 CRL 정보를 나타냅니다. 동적 CRL 정보는 인증서 체인에서 현재 처리 중인 인증서에 대해서만 처리됩니다. 정적 CRL은 전체 피어 인증서 체인에 적용됩니다.

보안 IPSec, LDAP 및 HTTPS 연결에 대한 인증서 해지 확인을 활성화 또는 비활성화하기 위한 단계에 대해서는 IPSec 보안 채널 구성, 66 페이지, LDAP 제공자 생성, 127 페이지 및 HTTPS 구성, 121 페이지의 내용을 참조하십시오.

참고

- Certificate Revocation Check Mode(인증서 해지 확인 모드)를 Strict(엄격)로 설정한 경우, 피어 인증서 체인이 레벨 1 이상인 경우에만 정적 CRL을 적용할 수 있습니다. 피어 인증서 체인이 루트 CA 인증서만 포함하고 피어 인증서가 루트 CA에 의해 서명된 경우를 예로 들 수 있습니다.
- IPSec에 대해 정적 CRL을 구성할 때 가져온 CRL 파일에 인증 키 식별자(authkey) 필드가 있어야 합니다. 그렇지 않으면 IPSec에서 해당 파일을 유효하지 않은 것으로 간주합니다.
- 정적 CRL은 동일한 발급자의 동적 CRL보다 우선적으로 적용됩니다. 피어 인증서를 검증할 때 동일한 발급자의 유효한(확인된) 정적 CRL이 있는 경우, 피어 인증서의 CDP는 무시됩니다.

다음 표에는 인증서 해지 목록 확인 설정 및 인증서 검증에 따른 연결 결과가 나와 있습니다.

표 4: 로컬 정적 CRL 없이 Strict(엄격)로 설정된 인증서 해지 확인 모드

로컬 정적 CRL 없음	LDAP 연결	IPSec 연결	클라이언트 인증서 인증
피어 인증서 체인 확인	전체 인증서 체인 필요	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인에서 CDP 확인	전체 인증서 체인 필요	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음	예
피어 인증서 체인에서 인증서 유효성 검사 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨	Syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 실패	Syslog 메시지와 함께 연결 실패
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공	Syslog 메시지와 함께 연결 실패

로컬 정적 CRL 없음	LDAP 연결	IPSec 연결	클라이언트 인증서 인증
피어 인증서 체인에서 모든 CDP를 다운로드할 수 없음	Syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 실패	Syslog 메시지와 함께 연결 실패
인증서에 CDP가 있지만 CDP 서버가 다운됨	Syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 실패	Syslog 메시지와 함께 연결 실패
인증서에 CDP가 있고 서버가 작동 중이며 CRL이 CDP에 있지만 CRL에 유효하지 않은 서명이 있음	Syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 실패	Syslog 메시지와 함께 연결 실패

표 5: 로컬 정적 CRL이 있으며 Strict(엄격)로 설정된 인증서 해지 확인 모드

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
피어 인증서 체인 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인에서 CDP 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨(인증서 체인 레벨이 1임)	연결 성공	연결 성공
피어 인증서 체인에서 하나의 CDP CRL이 비어 있음(인증서 체인 레벨이 1임)	연결 성공	연결 성공
피어 인증서 체인에서 모든 CDP를 다운로드할 수 없음(인증서 체인 레벨이 1임)	연결 성공	연결 성공

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
인증서에 CDP가 있지만 CDP 서버가 다운됨(인증서 체인 레벨이 1임)	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이며 CRL이 CDP에 있지만 CRL에 유효하지 않은 서명이 있음(인증서 체인 레벨이 1임)	연결 성공	연결 성공
피어 인증서 체인 레벨이 1보다 높음	Syslog 메시지와 함께 연결 실패	CDP와 결합된 경우 연결 성공 CDP가 없는 경우 syslog 메시지와 함께 연결 실패

표 6: 로컬 정적 CRL 없이 Relaxed(완화)로 설정된 인증서 해지 확인 모드

로컬 정적 CRL 없음	LDAP 연결	IPSec 연결	클라이언트 인증서 인증
피어 인증서 체인 확인	전체 인증서 체인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인에서 CDP 확인	전체 인증서 체인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음	예
피어 인증서 체인에서 인증서 유효성 검사 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨	연결 성공	연결 성공	Syslog 메시지와 함께 연결 실패
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공	연결 성공

로컬 정적 CRL 없음	LDAP 연결	IPSec 연결	클라이언트 인증서 인증
피어 인증서 체인에서 모든 CDP를 다운로드할 수 없음	연결 성공	연결 성공	연결 성공
인증서에 CDP가 있지만 CDP 서버가 다운됨	연결 성공	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이며 CRL이 CDP에 있지만 CRL에 유효하지 않은 서명이 있음	연결 성공	연결 성공	연결 성공

표 7: 로컬 정적 CRL이 있으며 Relaxed(완화)로 설정된 인증서 해지 확인 모드

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
피어 인증서 체인 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인에서 CDP 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	Syslog 메시지와 함께 연결 실패	Syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨(인증서 체인 레벨이 1임)	연결 성공	연결 성공
피어 인증서 체인에서 하나의 CDP CRL이 비어 있음(인증서 체인 레벨이 1임)	연결 성공	연결 성공
피어 인증서 체인에서 모든 CDP를 다운로드할 수 없음(인증서 체인 레벨이 1임)	연결 성공	연결 성공

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
인증서에 CDP가 있지만 CDP 서버가 다운됨(인증서 체인 레벨이 1임)	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이며 CRL이 CDP에 있지만 CRL에 유효하지 않은 서명이 있음(인증서 체인 레벨이 1임)	연결 성공	연결 성공
피어 인증서 체인 레벨이 1보다 높음	Syslog 메시지와 함께 연결 실패	CDP와 결합된 경우 연결 성공 CDP가 없는 경우 syslog 메시지와 함께 연결 실패

CRL의 주기적인 다운로드 구성

CRL을 주기적으로 다운로드하도록 시스템을 구성하여 1~24시간마다 새 CRL을 사용하여 인증서를 검증할 수 있습니다.

이 기능과 함께 다음 프로토콜 및 인터페이스를 사용할 수 있습니다.

- FTP
- SCP
- SFTP
- TFTP
- USB

참고
- SCEP 및 OCSP는 지원되지 않습니다.
- 주기적 다운로드는 CRL당 하나만 구성할 수 있습니다.
- 트러스트 포인트당 하나의 CRL이 지원됩니다.

참고 기간은 1시간 간격으로만 구성할 수 있습니다.

CRL 주기적 다운로드를 구성하려면 다음 단계를 수행하십시오.

시작하기 전에

CRL 정보를 사용하여 피어 인증서를 검증하려면 Firepower 4100/9300 섀시를 이미 구성했는지 확인합니다. 자세한 내용은 트러스트 포인트에 대한 정적 CRL 구성, 71 페이지를 참조하십시오.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesecurity

단계 **2** 트러스트 포인트 모드를 시작합니다.
scopetrustpoint

단계 **3** 취소 모드를 시작합니다.
scoperevoke

단계 **4** 취소 컨피그레이션을 편집합니다.
shconfig

단계 **5** 기본 컨피그레이션을 설정합니다.

예제:

```
set certrevokemethod crl
set crl-poll-filename rootCA.crl
set crl-poll-path /users/myname
set crl-poll-period 1
set crl-poll-port 0
set crl-poll-protocol scp
! set crl-poll-pwd
set crl-poll-server 182.23.33.113
set crl-poll-user myname
```

단계 **6** 컨피그레이션 파일을 종료합니다.
exit

단계 **7** (선택 사항) 새 CRL을 다운로드하여 새 컨피그레이션을 테스트합니다.

예제:

```
Firepower-chassis /security/trustpoint/revoke # sh import-task

Import task:
File Name Protocol Server          Port      Userid        State
--------- -------- --------------- ---------- -------------- -----
rootCA.crl Scp     182.23.33.113   0         myname        Downloading
```

NTP 서버 인증 활성화

Firepower 4100/9300 섀시에서 NTP 서버 인증을 활성화하려면 다음 단계를 수행합니다.

참고

- NTP 인증 기능은 활성화하는 경우, 구성된 모든 서버에 전역적으로 적용됩니다.
- NTP 서버 인증에는 SHA1만 지원됩니다.
- 서버 인증을 위해서는 키 ID와 키 값이 필요합니다. 키 ID는 메시지 다이제스트를 컴퓨팅할 때 어떤 키 값을 사용할지 클라이언트와 서버에 알리는 데 사용됩니다. 키 값은 ntp-keygen을 사용하여 파생된 고정 값입니다.

절차

단계 **1** ntp 4.2.8p8을 다운로드합니다.

단계 **2** **ntpd openssl**이 활성화되어 있는 NTP 서버를 설치합니다.

단계 **3** NTP 키 ID와 키 값을 생성합니다.
ntp-keygen-M

생성된 키를 사용하여 다음 단계를 수행합니다.

단계 **4** FXOS CLI에서 NTP 서버를 생성합니다.
createntp-server *server_id*

단계 **5** NTP 서버를 입력합니다.
scopentp-server *server_id*

단계 **6** SHA1 키 ID를 설정합니다.
setntp-sha1-key-id *key_id*

단계 **7** SHA1 키 문자열을 설정합니다.
setntp-sha1-key-string *key_string*

단계 **8** NTP 인증을 활성화합니다.
enablentp-authentication

LDAP 키 링 인증서 설정

Firepower 4100/9300 섀시에서 TLS 연결을 지원하도록 보안 LDAP 클라이언트 키 링 인증서를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증 컴플라이언스, 63 페이지를 참조하십시오.

참고 Common Criteria 모드가 활성화되면 SSL을 활성화하고, 서버 DNS 정보를 사용하여 키 링 인증서를 생성해야 합니다.

LDAP 서버 항목에 대해 SSL이 활성화되면 연결을 설정할 때 키 링 정보를 참조하고 확인해야 합니다.

LDAP 서버 정보는 보안 LDAP 연결(SSL이 활성화된 상태)에 대한 CC 모드에서 DNS 정보여야 합니다.

보안 LDAP 클라이언트 키 링 인증서를 구성하려면 다음 단계를 수행합니다.

절차

단계 **1** FXOS CLI에서 보안 모드를 시작합니다.
scopesecurity

단계 **2** LDAP 모드를 시작합니다.
scopeldap

단계 **3** LDAP 서버 모드를 시작합니다.
enterserver *{server_ip|server_dns}*

단계 **4** LDAP 키 링을 설정합니다.
setkeyring *keyring_name*

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

IP 액세스 목록 구성

기본적으로 Firepower 4100/9300 새시는 로컬 웹 서버에 대한 모든 액세스를 거부합니다. 각 IP 블록의 허용된 서비스 목록을 사용하여 IP 액세스 목록을 구성해야 합니다.

IP 액세스 목록은 다음 프로토콜을 지원합니다.

- HTTPS
- SNMP
- SSH

IP 주소(v4 또는 v6)의 각 블록의 경우, 각 서비스에 대해 최대 25개의 서로 다른 서브넷을 구성할 수 있습니다. 서브넷과 접두사가 모두 0인 경우 서비스에 무제한으로 액세스할 수 있습니다.

절차

단계 **1** FXOS CLI에서 서비스 모드를 시작합니다.
scopesystem

scopeservices

단계 **2** 액세스를 활성화할 서비스에 대해 IP 블록을 생성합니다.
IPv4의 경우:
createip-block *ip prefix [0-32]* [**http** | **snmp** | **ssh**]

IPv6의 경우:
createipv6-block *ip prefix [0-28]* [**http** | **snmp** | **ssh**]

IPv4:

```
Firepower-chassis # scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create ip-block 10.1.1.1 24 https
Firepower-chassis /system/services/ip-block* # com
Firepower-chassis /system/services/ip-block # up
Firepower-chassis /system/services # create ip-block 11.1.1.1 24 ssh
Firepower-chassis /system/services/ip-block* # com
Firepower-chassis /system/services/ip-block # up
Firepower-chassis /system/services # create ip-block 12.1.1.1 24 snmp
Firepower-chassis /system/services/ip-block* # com
Firepower-chassis /system/services/ip-block # up
Firepower-chassis /system/services # sh ip-block
Permitted IP Block:
    IP Address      Prefix Length Protocol
    --------------- ------------- --------
    10.1.1.1        24            Https
    11.1.1.1        24            Ssh
    12.1.1.1        24            Snmp
```

IPv6:

```
Firepower-chassis /system/services # create ipv6-block 2014::10:76:78:107 64 ssh
Firepower-chassis /system/services/ipv6-block* # com
Firepower-chassis /system/services/ipv6-block # up
Firepower-chassis /system/services # create ipv6-block 2014::10:76:78:107 64 snmp
Firepower-chassis /system/services/ipv6-block* # com
Firepower-chassis /system/services/ipv6-block # up
Firepower-chassis /system/services # create ipv6-block 2014::10:76:78:107 64 https
Firepower-chassis /system/services/ipv6-block* # com
Firepower-chassis /system/services/ipv6-block # up
Firepower-chassis /system/services # sh ipv6-block
Permitted IPv6 Block:
    IPv6 Address Prefix Length Protocol
    ------------ ------------- --------
    2014::10:76:78:107   64      Https
    2014::10:76:78:107   64      Snmp
    2014::10:76:78:107   64      Ssh
```

클라이언트 인증서 인증 활성화

LDAP와 함께 클라이언트 인증서를 사용하도록 시스템을 활성화하여 HTTPS 액세스에 대해 사용자를 인증할 수 있습니다. Firepower 4100/9300 섀시의 기본 인증 컨피그레이션은 자격 증명 기반입니다.

참고 인증서 인증이 활성화된 경우, 이 인증은 HTTPS에 대해 허용되는 유일한 인증 형식입니다.

인증서 해지 확인은 클라이언트 인증서 인증 기능의 FXOS 2.1.1 릴리스에 지원되지 않습니다.

이 기능을 사용하려면 클라이언트 인증서는 다음 요건을 충족해야 합니다.

- X509 속성 Subject Alternative Name - Email(주체 대체 이름 - 이메일)에 사용자 이름을 포함해야 합니다.
- 클라이언트 인증서에 해당 인증서를 수퍼바이저의 트러스트 포인트로 가져온 루트 CA의 서명이 있어야 합니다.

절차

단계 **1** FXOS CLI에서 서비스 모드를 시작합니다.
scopesystem

scopesecurity

단계 **2** (선택 사항) HTTPS 인증의 옵션을 확인합니다.
sethttpsauth-type

예제:
```
Firepower-chassis /system/services # set https auth-type
cert-auth Client certificate based authentication
cred-auth Credential based authentication
```

단계 **3** HTTPS 인증을 클라이언트 기반으로 설정합니다.
sethttpsauth-typecert-auth

단계 **4** 컨피그레이션을 커밋합니다.
commit-buffer

8 장

시스템 관리

관리 IP 주소 변경

시작하기 전에

Firepower 4100/9300 섀시의 관리 IP 주소를 FXOS CLI에서 변경할 수 있습니다.

참고 관리 IP 주소를 변경한 후, 새 주소를 사용하여 Firepower Chassis Manager 또는 FXOS CLI에 대한 모든 연결을 다시 설정해야 합니다.

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 다음과 같이 IPv4 관리 IP 주소를 구성합니다.

a) Fabric-interconnect a에 대한 범위를 설정합니다.
Firepower-chassis# **scopefabric-interconnecta**

b) 다음 명령을 입력하여 현재 관리 IP 주소를 확인합니다.

Firepower-chassis /fabric-interconnect # **show**

c) 다음 명령을 입력하여 새로운 관리 IP 주소 및 게이트웨이를 구성합니다.
Firepower-chassis /fabric-interconnect # **setout-of-band staticip** *ip_address***netmask** *network_mask***gw** *gateway_ip_address*

d) 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /fabric-interconnect* # **commit-buffer**

단계 **3** 다음과 같이 IPv6 관리 IP 주소를 구성합니다.

a) Fabric-interconnect a에 대한 범위를 설정합니다.
Firepower-chassis# **scopefabric-interconnecta**

b) 관리 IPv6 컨피그레이션의 범위를 설정합니다.
Firepower-chassis /fabric-interconnect # **scopeipv6-config**

c) 다음 명령을 입력하여 현재 관리 IPv6 주소를 확인합니다.
Firepower-chassis /fabric-interconnect/ipv6-config # **show ipv6-if**

d) 다음 명령을 입력하여 새로운 관리 IP 주소 및 게이트웨이를 구성합니다.
Firepower-chassis /fabric-interconnect/ipv6-config # **setout-of-band staticipv6** *ipv6_address***ipv6-prefix** *prefix_length***ipv6-gw** *gateway_address*

e) 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /fabric-interconnect/ipv6-config* # **commit-buffer**

다음 예에서는 IPv4 관리 인터페이스 및 게이트웨이를 구성합니다.

```
Firepower-chassis# scope fabric-interconnect a
Firepower-chassis /fabric-interconnect # show

Fabric Interconnect:
    ID   OOB IP Addr     OOB Gateway     OOB Netmask     OOB IPv6 Address OOB IPv6 Gateway
 Prefix Operability
    ---- --------------- --------------- --------------- ---------------- ----------------
 ------ -----------
    A    192.0.2.112     192.0.2.1       255.255.255.0   ::               ::
 64     Operable
Firepower-chassis /fabric-interconnect # set out-of-band static ip 192.0.2.111 netmask
255.255.255.0 gw 192.0.2.1
Warning: When committed, this change may disconnect the current CLI session
Firepower-chassis /fabric-interconnect* #commit-buffer
Firepower-chassis /fabric-interconnect #
```

다음 예에서는 IPv6 관리 인터페이스 및 게이트웨이를 구성합니다.

```
Firepower-chassis# scope fabric-interconnect a
Firepower-chassis /fabric-interconnect # scope ipv6-config
Firepower-chassis /fabric-interconnect/ipv6-config # show ipv6-if

Management IPv6 Interface:
    IPv6 Address                      Prefix     IPv6 Gateway
    --------------------------------- ---------- ------------
    2001::8998                        64         2001::1
Firepower-chassis /fabric-interconnect/ipv6-config # set out-of-band static ipv6 2001::8999
 ipv6-prefix 64 ipv6-gw 2001::1
Firepower-chassis /fabric-interconnect/ipv6-config* # commit-buffer
Firepower-chassis /fabric-interconnect/ipv6-config #
```

애플리케이션 관리 IP 변경

Firepower 4100/9300 섀시에 연결된 애플리케이션의 관리 IP 주소를 FXOS CLI에서 변경할 수 있습니다. 이렇게 하려면 먼저 FXOS 플랫폼 레벨에서 IP 정보를 변경한 다음 애플리케이션 레벨에서 IP 정보를 변경해야 합니다.

참고 Firepower Chassis Manager를 사용하여 이러한 변경을 수행하려고 시도하면 서비스가 중단될 수 있습니다. 잠재적 서비스 중단을 피하려면 FXOS CLI를 사용해 이러한 변경을 수행해야 합니다.

절차

단계 **1** FXOS CLI에 연결합니다. (FXOS CLI 액세스, 14 페이지를 참조하십시오.)

단계 **2** 논리적 디바이스로 범위를 한정합니다.
scopessa

scopelogical-device *logical_device_name*

단계 **3** 관리 부트스트랩으로 범위를 한정하고 새로운 관리 부트스트랩 파라미터를 구성합니다. 구축 간에는 차이점이 있습니다.
ASA 논리적 디바이스의 독립형 컨피그레이션의 경우:

a) 논리적 디바이스 관리 부트스트랩을 입력합니다.
scopemgmt-bootstrap *asa*

b) 슬롯의 IP 모드를 시작합니다.
scope *ipv4_or_6 slot_number* default

c) (IPv4만 해당) 새 IP 주소를 설정합니다.
setip *ipv4_address***mask** *network_mask*

d) (IPv6만 해당) 새 IP 주소를 설정합니다.
setip *ipv6_address***prefix-length** *prefix_length_number*

e) 게이트웨이 주소를 설정합니다.
setgateway *gateway_ip_address*

f) 컨피그레이션을 커밋합니다.
commit-buffer

ASA 논리적 디바이스의 클러스터형 컨피그레이션의 경우:

a) 클러스터 관리 부트스트랩을 입력합니다.
scopecluster-bootstrap *asa*

b) (IPv4만 해당) 새 가상 IP를 설정합니다.
setvirtualipv4 *ip_address***mask** *network_mask*

c) (IPv6만 해당) 새 가상 IP를 설정합니다.
setvirtualipv6 *ipv6_address***prefix-length** *prefix_length_number*

d) 새 IP 풀을 설정합니다.
setippool *start_ip end_ip*

e) 게이트웨이 주소를 설정합니다.
setgateway *gateway_ip_address*

f) 컨피그레이션을 커밋합니다.
commit-buffer

Firepower Threat Defense의 독립형 및 클러스터형 컨피그레이션의 경우:

a) 논리적 디바이스 관리 부트스트랩을 입력합니다.
scopemgmt-bootstrap *ftd*

b) 슬롯의 IP 모드를 시작합니다.
scope *ipv4_or_6 slot_number* firepower

c) (IPv4만 해당) 새 IP 주소를 설정합니다.
setip *ipv4_address***mask** *network_mask*

d) (IPv6만 해당) 새 IP 주소를 설정합니다.
setip *ipv6_address***prefix-length** *prefix_length_number*

e) 게이트웨이 주소를 설정합니다.
setgateway *gateway_ip_address*

f) 컨피그레이션을 커밋합니다.
commit-buffer

참고 클러스터형 컨피그레이션의 경우, Firepower 4100/9300 섀시에 연결된 각 애플리케이션의 새 IP 주소를 설정해야 합니다. 섀시 간 클러스터 또는 HA 컨피그레이션을 보유한 경우, 두 섀시 모두에서 각 애플리케이션에 대해 이 단계를 반복해야 합니다.

단계 **4** 각 애플리케이션에 대한 관리 부트스트랩 정보를 지웁니다.

a) ssa 모드로 범위를 한정합니다.
scopessa

b) 슬롯으로 범위를 한정합니다.
scopeslot *slot_number*

c) 애플리케이션 인스턴스로 범위를 한정합니다.
scopeapp-instance *asa_or_ftd*

d) 관리 부트스트랩 정보를 지웁니다.
clearmgmt-bootstrap

e) 컨피그레이션을 커밋합니다.
commit-buffer

단계 **5** 애플리케이션을 비활성화합니다.
disable

commit-buffer

참고 클러스터형 컨피그레이션의 경우, Firepower 4100/9300 섀시에 연결된 각 애플리케이션의 관리 부트스트랩 정보를 지우고 비활성화해야 합니다. 섀시 간 클러스터 또는 HA 컨피그레이션을 보유한 경우, 두 섀시 모두에서 각 애플리케이션에 대해 이 단계를 반복해야 합니다.

단계 **6** 애플리케이션이 오프라인 상태인 경우 슬롯이 다시 온라인 상태가 되면 애플리케이션을 다시 활성화합니다.

a) 다시 ssa 모드로 범위를 한정합니다.
scopessa

b) 슬롯으로 범위를 한정합니다.
scopeslot *slot_number*

c) 애플리케이션 인스턴스로 범위를 한정합니다.
scopeapp-instance *asa_or_ftd*

d) 애플리케이션을 활성화합니다.
enable

e) 컨피그레이션을 커밋합니다.
commit-buffer

참고 클러스터형 컨피그레이션의 경우, Firepower 4100/9300 섀시에 연결된 각 애플리케이션을 다시 활성화하려면 이 단계를 반복해야 합니다. 섀시 간 클러스터 또는 HA 컨피그레이션을 보유한 경우, 두 섀시 모두에서 각 애플리케이션에 대해 이 단계를 반복해야 합니다.

Firepower 4100/9300 섀시 이름 변경

시작하기 전에

Firepower 4100/9300 섀시에 사용되는 이름을 FXOS CLI에서 변경할 수 있습니다.

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 시스템 모드를 시작합니다.
Firepower-chassis-A# **scopesystem**

단계 **3** 현재 이름을 확인합니다.
Firepower-chassis-A /system # **show**

단계 **4** 새 이름을 구성합니다.
Firepower-chassis-A /system # **setname** *device_name*

단계 **5** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis-A /fabric-interconnect* # **commit-buffer**

다음 예에서는 디바이스 이름을 변경합니다.

```
Firepower-chassis-A# scope system
Firepower-chassis-A /system # set name New-name
```

```
Warning: System name modification changes FC zone name and redeploys them non-disruptively
Firepower-chassis-A /system* # commit-buffer
Firepower-chassis-A /system # show

Systems:
    Name       Mode        System IP Address System IPv6 Address
    ---------- ----------- ----------------- -------------------
    New-name   Stand Alone 192.168.100.10    ::
New-name-A /system #
```

사전 로그인 배너

사전 로그인 배너를 사용하면 사용자가 Firepower Chassis Manager에 로그인할 때 배너 텍스트가 표시되며, 사용자는 사용자 이름 및 비밀번호를 묻는 프롬프트가 표시되기 전에 메시지 화면에서 **OK(**확인**)**를 클릭해야 합니다. 사전 로그인 배너가 구성되어 있지 않은 경우에는 바로 사용자 이름 및 비밀번호 프롬프트가 표시됩니다.

구성되어 있는 경우에는 사용자가 FXOS CLI에 로그인하면 비밀번호를 묻는 프롬프트가 표시되기 전에 배너 텍스트가 표시됩니다.

사전 로그인 배너 생성

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 보안 모드를 시작합니다.
Firepower-chassis# **scopesecurity**

단계 **3** 배너 보안 모드를 시작합니다.
Firepower-chassis /security # **scopebanner**

단계 **4** 다음 명령을 입력하여 사전 로그인 배너를 생성합니다.
Firepower-chassis /security/banner # **create pre-login-banner**

단계 **5** 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하기 전에 FXOS에서 사용자에게 표시해야 할 메시지를 지정합니다.
Firepower-chassis /security/banner/pre-login-banner* # **set message**

사전 로그인 배너 메시지 텍스트를 입력할 대화 상자를 실행합니다.

단계 **6** 프롬프트에서 사전 로그인 배너 메시지를 입력합니다. 이 필드에 임의의 표준 ASCII 문자를 입력할 수 있습니다. 각 라인에 최대 192자를 포함할 수 있는 여러 라인의 텍스트를 입력할 수 있습니다. 각 라인 사이에서 **Enter** 키를 누릅니다.
입력한 후 라인에서 ENDOFBUF를 입력하고 완료하려면 **Enter** 키를 누릅니다.

설정 메시지 대화 상자를 취소하려면 Ctrl+C를 누릅니다.

단계 **7** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

Firepower-chassis /security/banner/pre-login-banner* # **commit-buffer**

다음 예에서는 사전 로그인 배너를 생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope banner
Firepower-chassis /security/banner # create pre-login-banner
Firepower-chassis /security/banner/pre-login-banner* # set message
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Enter prelogin banner:
>Welcome to the Firepower Security Appliance
>**Unauthorized use is prohibited**
>ENDOFBUF
Firepower-chassis /security/banner/pre-login-banner* # commit-buffer
Firepower-chassis /security/banner/pre-login-banner #
```

사전 로그인 배너 수정

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 보안 모드를 시작합니다.
Firepower-chassis# **scopesecurity**

단계 **3** 배너 보안 모드를 시작합니다.
Firepower-chassis /security # **scopebanner**

단계 **4** pre-login-banner 배너 보안 모드를 시작합니다.
Firepower-chassis /security/banner # **scope pre-login-banner**

단계 **5** 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하기 전에 FXOS에서 사용자에게 표시해야 할 메시지를 지정합니다.
Firepower-chassis /security/banner/pre-login-banner # **set message**

사전 로그인 배너 메시지 텍스트를 입력할 대화 상자를 실행합니다.

단계 **6** 프롬프트에서 사전 로그인 배너 메시지를 입력합니다. 이 필드에 임의의 표준 ASCII 문자를 입력할 수 있습니다. 각 라인에 최대 192자를 포함할 수 있는 여러 라인의 텍스트를 입력할 수 있습니다. 각 라인 사이에서 **Enter** 키를 누릅니다.
입력 다음 줄에 ENDOFBUF를 입력하고 **Enter**를 눌러 완료합니다.

설정 메시지 대화 상자를 취소하려면 Ctrl+C를 누릅니다.

단계 **7** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/banner/pre-login-banner* # **commit-buffer**

다음 예에서는 사전 로그인 배너를 수정합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope banner
Firepower-chassis /security/banner # scope pre-login-banner
```

```
Firepower-chassis /security/banner/pre-login-banner # set message
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Enter prelogin banner:
>Welcome to the Firepower Security Appliance
>**Unauthorized use is prohibited**
>ENDOFBUF
Firepower-chassis /security/banner/pre-login-banner* # commit-buffer
Firepower-chassis /security/banner/pre-login-banner #
```

사전 로그인 배너 삭제

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 보안 모드를 시작합니다.
Firepower-chassis# **scopesecurity**

단계 **3** 배너 보안 모드를 시작합니다.
Firepower-chassis /security # **scopebanner**

단계 **4** 시스템에서 사전 로그인 배너를 삭제합니다.
Firepower-chassis /security/banner # **delete pre-login-banner**

단계 **5** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/banner* # **commit-buffer**

다음 예에서는 사전 로그인 배너를 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope banner
Firepower-chassis /security/banner # delete pre-login-banner
Firepower-chassis /security/banner* # commit-buffer
Firepower-chassis /security/banner #
```

Firepower 4100/9300 섀시 재부팅

절차

단계 **1** 섀시 모드를 시작합니다.
scope chassis 1

단계 **2** 다음 명령을 입력하여 섀시를 재부팅합니다.
reboot [*reason*] **[no-prompt]**

참고 **[no-prompt]** 키워드를 사용하는 경우, 명령을 입력한 후 즉시 섀시가 재부팅됩니다. **[no-prompt]** 키워드를 사용하지 않는 경우, **commit-buffer** 명령을 입력할 때까지 시스템이 재부팅되지 않습니다.

시스템에 구성되어 있는 모든 논리적 디바이스를 정상적으로 종료한 다음 최종적으로 Firepower 4100/9300 섀시의 전원을 끄고 재시작하기 전에 각 보안 모듈/엔진의 전원을 끕니다. 이 프로세스에는 약 15~20분이 소요됩니다.

단계 **3** 다음 명령을 사용하여 재부팅 프로세스를 모니터링합니다.
scope chassis 1
show fsm status

Firepower 4100/9300 섀시 전원 끄기

절차

단계 **1** 섀시 모드를 시작합니다.
scope chassis 1

단계 **2** 다음 명령을 입력하여 섀시의 전원을 끕니다.
shutdown [*reason*] **[no-prompt]**

참고 **[no-prompt]** 키워드를 사용하는 경우, 명령을 입력한 후 즉시 섀시가 종료됩니다. **[no-prompt]** 키워드를 사용하지 않는 경우, **commit-buffer** 명령을 입력할 때까지 시스템이 종료되지 않습니다.

시스템에 구성되어 있는 모든 논리적 디바이스를 정상적으로 종료한 다음 최종적으로 Firepower 4100/9300 섀시의 전원을 끄기 전에 각 보안 모듈/엔진의 전원을 끕니다. 이 프로세스에는 약 15~20분이 소요됩니다. 섀시가 성공적으로 종료되어야 섀시에서 전원 플러그를 물리적으로 분리할 수 있습니다.

단계 **3** 다음 명령을 사용하여 종료 프로세스를 모니터링합니다.
scope chassis 1
show fsm status

신뢰할 수 있는 ID 인증서 설치

초기 컨피그레이션 이후 Firepower 4100/9300 섀시 웹 애플리케이션에서 사용하기 위한 자체 서명 SSL 인증서가 생성됩니다. 인증서가 자체 서명되므로 클라이언트 브라우저는 이 인증서를 자동으로 신뢰하지 않습니다. 새 클라이언트 브라우저가 처음으로 Firepower 4100/9300 섀시 웹 인터페이스에 액세스할 때 브라우저는 SSL 경고를 발생시켜 사용자에게 Firepower 4100/9300 섀시에 액세스하기 전에 인증서를 수락하도록 요청합니다. FXOS CLI를 사용하여 CSR(인증서 서명 요청)을 생성하고 Firepower 4100/9300 섀시에서 사용할 결과 ID 인증서를 설치하려면 다음 절차를 사용할 수 있습니다. 이 ID 인증서를 사용하면 클라이언트 브라우저에서 연결을 신뢰할 수 있으며 경고 없이 웹 인터페이스를 불러올 수 있습니다.

절차

단계 1 FXOS CLI에 연결합니다. (FXOS CLI 액세스, 14 페이지를 참조하십시오.)

단계 2 보안 모듈을 입력합니다.
scopesecurity

단계 3 키 링을 생성합니다.
createkeyring *keyring_name*

단계 4 개인 키에 대한 모듈러스 크기를 설정합니다.
setmodulus *size*

단계 5 컨피그레이션을 커밋합니다.
commit-buffer

단계 6 CSR 필드를 구성합니다. 인증서는 기본 옵션(예: 주제-이름)을 사용하여 생성할 수 있으며, 선택적으로 로캘 및 조직과 같은 정보가 인증서에 포함되도록 허용하는 고급 옵션을 사용할 수 있습니다. CSR 필드를 구성할 때 시스템에서 인증서 비밀번호를 묻습니다.
createcertreqcertreq *subject_name*

password

setcountry *country*

setstate *state*

setlocality *locality*

setorg-name *organization_name*

setorg-unit-name *organization_unit_name*

setsubject-name *subject_name*

단계 7 컨피그레이션을 커밋합니다.
commit-buffer

단계 8 인증 기관(CA)에 제공할 CSR을 내보냅니다.

a) 전체 CSR을 표시합니다.
showcertreq

b) "-----BEGIN CERTIFICATE REQUEST-----"로 시작되고 "-----END CERTIFICATE REQUEST-----"로 종료되는 출력을 복사합니다.

예제:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC6zCCAdMCAQAwdzELMAkGA1UEBhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWEx
ETAPBgNVBAcMCFNhbiBKb3NlMRYwFAYDVQQKDA1DaXNjbyBTeXN0ZW1zMQwwCgYD
VQQLDANUQUMxGjAYBgNVBAMMEWZwNDEyMC50ZXN0LmxvY2FsMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs0ON5gagkfZ2fi4JVEANG+7YGgcHbnUt7LpV
yMChnKOPJjBwkUMNQA1mQsRQDcbJ232/sK0fMSnyqOL8JzC7itxeVEZRyz7/ax7W
GNveg/XP+zd03nt4GXM63FsrPcPmA7EwgqDSLoShtBEV10hhf4+Nw4pKCZ+eSSkS
JkTB1ZHaKV9bttYg3kf/UEUUgk/EyrVq3B+u2DsooPVq76mTm8BwYMqHbJEv4Pmu
RjWE88yEvVwH7JTEij9OvxbatjDjVSJHZBURtCanvyBvGuLP/Q/Nmv3Lo3G9ITbL
L5gIYZVatTxp6HTUezH2MIIzOavU6d1tB9rnyxgGth5dPV0dhQIDAQABoC8wLQYJ
KoZIhvcNAQkOMSAwHjAcBgNVHREEFTATghFmcDQxMjAudGVzdC5sb2NhbDANBgkq
hkiG9w0BAQsFAAOCAQEAZUfCbwx9vt5aVDcL+tATu5xFE3LA310ck6GjlNv6W/6r
```

```
jBNLxusYi1rZZcW+CgnvNs4ArqYGyNVBySOavJO/VvQ1KfyxxJ1OIkyx3RzEjgK0
zzyoyrG+EZXC5ShiraS8HuWvE2wFM2wwWNtHWtvcQy55+/hDPD2Bv8pQOC2Zng3I
kLfG1dxWf1xAxLzf5J+AuIQ0CM5HzM9Zm8zREoWT+xHtLSqAqg/aCuomN9/vEwyU
OYfoJMvAqC6AZyUnMfUfCoyuLpLwgkxB0gyaRdnea5RhiGjYQ2lDXYDjEXp7rCx9
+6bvDl1n70JCegHdCWtP75SaNyaBEPkO0365rTckbw==
-----END CERTIFICATE REQUEST-----
```

단계 **9** certreq 모드를 종료합니다.
exit

단계 **10** 키 링 모드를 종료합니다.
exit

단계 **11** **참고** FXOS로 가져오려면 모든 인증서는 Base64 형식이어야 합니다. 인증 기관(CA)에서 받은 인증서 또는 체인이 다른 형식인 경우 먼저 OpenSSL과 같은 SSL 툴로 변환해야 합니다.인증서 체인을 보관하기 위해 새 트러스트 포인트를 생성합니다.

createtrustpoint *trustpoint_name*

단계 **12** 트러스트 포인트에서 생성된 CSR을 설정합니다.
setcertchain

단계 **13** **참고** 중간 인증서를 사용하는 인증 기관의 경우 루트 및 중간 인증서를 결합해야 합니다. 텍스트 파일에서 맨 위에 루트 인증서를 붙여넣고, 그 뒤에 체인의 각 중간 인증서를 붙여넣습니다(모든 BEGIN CERTIFICATE 및 END CERTIFICATE 플래그 포함). 해당하는 전체 텍스트 블록을 트러스트 포인트에 복사하여 붙여넣습니다.

화면의 지침에 따라 8단계에서 복사한 CSR 출력을 입력합니다.

예제:

```
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
>-----BEGIN CERTIFICATE-----
>MIICDTCCAbOgAwIBAgIQYIutxPDPw6BOp3uKNgJHZDAKBggqhkjOPQQDAjBTMRUw
>EwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJkiaJk/IsZAEZFghuYWF1c3RpbjEg
>MB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4tUEMtQ0EwHhcNMTUwNzI4MTc1NjU2
>WhcNMjAwNzI4MTgwNjU2WjBTMRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJ
>kiaJk/IsZAEZFghuYWF1c3RpbjEgMB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4t
>UEMtQ0EwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAASvEA27V1Enq1gMtLkvJ6rx
>GXRpXWIEyuiBM4eQRoqZKnkeJUkm1xmqlubaDHPJ5TMGfJQYszLBRJPq+mdrKcDl
>o2kwZzATBgkrBgEEAYI3FAIEBh4EAEMAQTAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0T
>AQH/BAUwAwEB/zAdBgNVHQ4EFgQUyInbDHPrFwEEBcbxGSgQW7pOVIkwEAYJKwYB
>BAGCNxUBBAMCAQAwCgYIKoZIzj0EAwIDSAAwRQIhAP++QJTUmniB/AxPDDN63Lqy
>18odMDoFTkG4p3Tb/2yMAiAtMYhlsv1gCxsQVOw0xZVRugSdoOak6n7wCjTFX9jr
>RA==
>-----END CERTIFICATE-----
>ENDOFBUF
```

단계 **14** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **15** 트러스트 포인트 모드를 종료합니다.
exit

단계 **16** 키 링 모드를 시작합니다.
scopekeyring *keyring_name*

단계 **17** 13단계에서 생성한 트러스트 포인트와 CSR용으로 생성된 키 링을 연결합니다.
settrustpoint *trustpoint_name*

단계 **18** 서버용으로 서명된 ID 인증서를 가져옵니다.
setcert

단계 **19** 인증 기관에서 제공하는 ID 인증서의 내용을 붙여넣습니다.

예제:

```
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
>-----BEGIN CERTIFICATE-----
>MIIE8DCCBJagAwIBAgITRQAAAArehlUWgiTzvgAAAAAACjAKBggqhkjOPQQDAjBT
>MRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJkiaJk/IsZAEZFghuYWF1c3Rp
>bjEgMB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4tUEMtQ0EwHhcNMTYwNDI4MTMw
>OTU0WhcNMTgwNDI4MTMwOTU0WjB3MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2Fs
>aWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxFjAUBgNVBAoTDUNpc2NvIFN5c3Rl
>bXMxDDAKBgNVBAsTA1RBQzEaMBgGA1UEAxMRZnA0MTIwLnRlc3QubG9jYWwwggEi
>MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCzQ43mBqCR9nZ+LglUQA0b7tga
>BwdudS3sulXIwKGco48mMHCRQwlADWZCxFANxsnbfb+wrR8xKfKo4vwnMLuK3F5U
>RlHLPv9rHtYY296D9c/7N3Tee3gZczrcWys9w+YDsTCCoNIuhKG0ERXXSGF/j43D
>ikoJn55JKRImRMHVkdopX1u21iDeR/9QRRSCT8TKtWrcH67YOyig9WrvqZObwHBg
>yodskS/g+a5GNYTzzIS9XAfslMSKP06/Ftq2MONVIkdkFRG0Jqe/IG8a4s/9D82a
>/cujcb0hNssvmAhhlVq1PGnodNR7MfYwgjM5q9Tp3W0H2ufLGAa2Hl09XR2FAgMB
>AAGjggJYMIICVDAcBgNVHREEFTATghFmcDQxMjAudGVzdC5sb2NhbDAdBgNVHQ4E
>FgQU/1WpstiEYExs8DlZWcuHZwPtu5QwHwYDVR0jBBgwFoAUyInbDHPrFwEEBcbx
>GSgQW7pOVIkwgdwGA1UdHwSB1DCB0TCBzqCBy6CByIaBxWxkYXA6Ly8vQ049bmFh
>dXN0aW4tTkFBVVNUSU4tUEMtQ0EsQ049bmFhdXN0aW4tcGMsQ049Q0RQLENOPVB1
>YmxpYyUyMEtleSUyMFNlcnZpY2VzLENOPVNlcnZpY2VzLENOPUNvbmZpZ3VyYXRp
>b24sREM9bmFhdXN0aW4sREM9bG9jYWw/Y2VydGlmaWNhdGVSZXZvY2F0aW9uTGlz
>dD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50MIHMBggrBgEF
>BQcBAQSBvzCBvDCBuQYIKwYBBQUHMAKGgaxsZGFwOi8vL0NOPW5hYXVzdGluLU5B
>QVVTVElOLVBDLUNBLENOPUFJQSxDTj1QdWJsaWMlMjBLZXklMjBTZXJ2aWNlcyxD
>Tj1TZXJ2aWNlcyxDTj1Db25maWd1cmF0aW9uLERDPW5hYXVzdGluLERDPWxvY2Fs
>P2NBQ2VydGlmaWNhdGU/YmFzZT9vYmplY3RDbGFzcz1jZXJ0aWZpY2F0aW9uQXV0
>aG9yaXR5MCEGCSsGAQQBgjcUAgQUHhIAVwBlAGIAUwBlAHIAdgBlAHIwDgYDVR0P
>AQH/BAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAoGCCqGSM49BAMCA0gAMEUC
>IFew7NcJirEtFRvyxjkQ4/dVo2oI6CRB308WQbYHNUu/AiEA7UdObiSJBG/PBZjm
>sgoIK60akbjotOTvUdUd9b6K1Uw=
>-----END CERTIFICATE-----
>ENDOFBUF
```

단계 **20** 키 링 모드를 종료합니다.
exit

단계 **21** 보안 모드를 종료합니다.
exit

단계 **22** 시스템 모드를 시작합니다.
scopesystem

단계 **23** 서비스 모드를 시작합니다.
scopeservices

단계 **24** 새 인증서를 사용하도록 FXOS 웹 서비스를 구성합니다.
sethttpskeyring *keyring_name*

단계 **25** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **26** HTTPS 서버와 연결된 키 링을 표시합니다. 여기에는 이 절차의 3단계에서 생성한 키 링 이름이 반영되어야 합니다. 화면 출력에 기본 키 링 이름이 표시되면 HTTPS 서버가 아직 새 인증서를 사용하도록 업데이트되지 않은 것입니다.

showhttps

예제:

```
fp4120 /system/services # show https
Name: https
    Admin State: Enabled
    Port: 443
    Operational port: 443
    Key Ring: firepower_cert
    Cipher suite mode: Medium Strength
   Cipher suite: ALL:!ADH:!EXPORT40:!EXPORT56:!LOW:!RC4:!MD5:!IDEA:+HIGH:+MEDIUM:+EXP:+eNULL
```

단계 **27** 가져온 인증서의 내용을 표시하고 해당 **Certificate Status**(인증서 상태) 값이 **Valid**(유효함)로 표시되는지 확인합니다.

scopesecurity

showkeyring *keyring_name***detail**

예제:

```
fp4120 /security #  scope security
fp4120 /security #  show keyring firepower_cert detail
Keyring firepower_cert:
    RSA key modulus: Mod2048
    Trustpoint CA: firepower_chain
    Certificate status: Valid
    Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            45:00:00:00:0a:de:86:55:16:82:24:f3:be:00:00:00:00:00:0a
    Signature Algorithm: ecdsa-with-SHA256
        Issuer: DC=local, DC=naaustin, CN=naaustin-NAAUSTIN-PC-CA
        Validity
            Not Before: Apr 28 13:09:54 2016 GMT
            Not After : Apr 28 13:09:54 2018 GMT
        Subject: C=US, ST=California, L=San Jose, O=Cisco Systems, OU=TAC,
CN=fp4120.test.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b3:43:8d:e6:06:a0:91:f6:76:7e:2e:09:54:40:
                    0d:1b:ee:d8:1a:07:07:6e:75:2d:ec:ba:55:c8:c0:
                    a1:9c:a3:8f:26:30:70:91:43:0d:40:0d:66:42:c4:
                    50:0d:c6:c9:db:7d:bf:b0:ad:1f:31:29:f2:a8:e2:
                    fc:27:30:bb:8a:dc:5e:54:46:51:cb:3e:ff:6b:1e:
                    d6:18:db:de:83:f5:cf:fb:37:74:de:7b:78:19:73:
                    3a:dc:5b:2b:3d:c3:e6:03:b1:30:82:a0:d2:2e:84:
                    a1:b4:11:15:d7:48:61:7f:8f:8d:c3:8a:4a:09:9f:
                    9e:49:29:12:26:44:c1:d5:91:da:29:5f:5b:b6:d6:
                    20:de:47:ff:50:45:14:82:4f:c4:ca:b5:6a:dc:1f:
                    ae:d8:3b:28:a0:f5:6a:ef:a9:93:9b:c0:70:60:ca:
                    87:6c:91:2f:e0:f9:ae:46:35:84:f3:cc:84:bd:5c:
                    07:ec:94:c4:8a:3f:4e:bf:16:da:b6:30:e3:55:22:
                    47:64:15:11:b4:26:a7:bf:20:6f:1a:e2:cf:fd:0f:
                    cd:9a:fd:cb:a3:71:bd:21:36:cb:2f:98:08:61:95:
                    5a:b5:3c:69:e8:74:d4:7b:31:f6:30:82:33:39:ab:
                    d4:e9:dd:6d:07:da:e7:cb:18:06:b6:1e:5d:3d:5d:
                    1d:85
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:fp4120.test.local
            X509v3 Subject Key Identifier:
                FF:55:A9:B2:D8:84:60:4C:6C:F0:39:59:59:CB:87:67:03:ED:BB:94
```

```
            X509v3 Authority Key Identifier:
                keyid:C8:89:DB:0C:73:EB:17:01:04:05:C6:F1:19:28:10:5B:BA:4E:54:89
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:ldap:///CN=naaustin-NAAUSTIN-PC-CA,CN=naaustin-pc,CN=CDP,
                    CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=naaustin,
                  DC=local?certificateRevocationList?base?objectClass=cRLDistributionPoint

            Authority Information Access:
                CA Issuers - URI:ldap:///CN=naaustin-NAAUSTIN-PC-CA,CN=AIA,
                  CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=naaustin,
                  DC=local?cACertificate?base?objectClass=certificationAuthority
            1.3.6.1.4.1.311.20.2:
                ...W.e.b.S.e.r.v.e.r
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:45:02:20:57:b0:ec:d7:09:8a:b1:2d:15:1b:f2:c6:39:10:
         e3:f7:55:a3:6a:08:e8:24:41:df:4f:16:41:b6:07:35:4b:bf:
         02:21:00:ed:47:4e:6e:24:89:04:6f:cf:05:98:e6:b2:0a:08:
         2b:ad:1a:91:b8:e8:b4:e4:ef:51:d5:1d:f5:be:8a:d5:4c
-----BEGIN CERTIFICATE-----
MIIE8DCCBJagAwIBAgITRQAAAArehlUWgiTzvgAAAAACjAKBggqhkjOPQQDAjBT
MRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJkiaJk/IsZAEZFghuYWFlc3Rp
bjEgMB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4tUEMtQ0EwHhcNMTYwNDI4MTMw
OTU0WhcNMTgwNDI4MTMwOTU0WjB3MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2Fs
aWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxFjAUBgNVBAoTDUNpc2NvIFN5c3Rl
bXMxDDAKBgNVBAsTA1RBQzEaMBgGA1UEAxMRZnA0MTIwLnRlc3QubG9jYWwwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCzQ43mBqCR9nZ+LglUQA0b7tga
BwdudS3sulXIwKGco48mMHCRQw1ADWZCxFANxsnbfb+wrR8xKfKo4vwnMLuK3F5U
RlHLPv9rHtYY296D9c/7N3Tee3gZczrcWys9w+YDsTCCoNIuhKG0ERXXSGF/j43D
ikoJn55JKRImRMHVkdopX1u21iDeR/9QRRSCT8TKtWrcH67YOyig9WrvqZObwHBg
yodskS/g+a5GNYTzzIS9XAfslMSKP06/Ftq2MONVIkdkFRG0Jqe/IG8a4s/9D82a
/cujcb0hNssvmAhhlVq1PGnodNR7MfYwgjM5q9Tp3W0H2ufLGAa2Hl09XR2FAgMB
AAGjggJYMIICVDAcBgNVHREEFTATghFmcDQxMjAudGVzdC5sb2NhbDAdBgNVHQ4E
FgQU/1WpstiEYExs8DlZWcuHZwPtu5QwHwYDVR0jBBgwFoAUyInbDHPrFwEEBcbx
GSgQW7pOVIkwgdwGA1UdHwSB1DCB0TCBzqCBy6CByIaBxWxkYXA6Ly8vQ049bmFh
dXN0aW4tTkFBVVNUSU4tUEMtQ0EsQ049bmFhdXN0aW4tcGMsQ049Q0RQLENOPVB1
YmxpYyUyMEtleSUyMFNlcnZpY2VzLENOPVNlcnZpY2VzLENOPUNvbmZpZ3VyYXRp
b24sREM9bmFhdXN0aW4sREM9bG9jYWw/Y2VydGlmaWNhdGVSZXZvY2F0aW9uTGlz
dD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50MIHMBggrBgEF
BQcBAQSBvzCBvDCBuQYIKwYBBQUHMAKGgaxsZGFwOi8vL0NOPW5hYXVzdGluLU5B
QVVTVElOLVBDLUNBLENOPUFJQSxDTj1QdWJsaWMlMjBLZXklMjBTZXJ2aWNlcyxD
Tj1TZXJ2aWNlcyxDTj1Db25maWd1cmF0aW9uLERDPW5hYXVzdGluLERDPWxvY2Fs
P2NBQ2VydGlmaWNhdGU/YmFzZT9vYmplY3RDbGFzcz1jZXJ0aWZpY2F0aW9uQXV0
aG9yaXR5MCEGCSsGAQQBgjcUAgQUHhIAVwBlAGIAUwBlAHIAdgBlAHIwDgYDVR0P
AQH/BAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAoGCCqGSM49BAMCA0gAMEUC
IFew7NcJirEtFRvyxjkQ4/dVo2oI6CRB308WQbYHNUu/AiEA7UdObiSJBG/PBZjm
sgoIK60akbjotOTvUdUd9b6K1Uw=
-----END CERTIFICATE-----

    Zeroized: No
```

다음에 할 작업

신뢰할 수 있는 새 인증서가 제공되는지 확인하려면 웹 브라우저의 주소 표시줄에 *https://<FQDN_or_IP>/*를 입력하여 Firepower Chassis Manager로 이동합니다.

참고 브라우저는 또한 주소 표시줄의 입력을 기준으로 인증서의 subject-name을 확인합니다. 인증서가 FQDN(Fully Qualified Domain Name)으로 발급된 경우 브라우저에서 해당 방식으로 액세스해야 합니다. IP 주소를 통해 액세스하는 경우, 신뢰할 수 있는 인증서가 사용되더라도 다른 SSL 오류가 표시됩니다(Common Name Invalid).

9 장

플랫폼 설정

날짜 및 시간 설정

날짜 및 시간을 수동으로 설정하거나 현재 시스템 시간을 보려면 아래에 설명된 CLI 명령을 사용하여 시스템에서 NTP(Network Time Protocol)를 구성합니다.

NTP 설정은 Firepower 4100/9300 섀시와 섀시에 설치된 논리적 디바이스 간에 자동으로 동기화됩니다.

참고 Firepower 4100/9300 섀시에서 Firepower Threat Defense를 구축 중인 경우, Firepower 4100/9300 섀시에 NTP를 구성해야 Smart Licensing이 제대로 작동하고 디바이스 등록 시 적절한 타임스탬프를 보장할 수 있습니다. Firepower 4100/9300 섀시와 Firepower Management Center에는 동일한 NTP 서버를 사용해야 합니다.

NTP를 사용 중인 경우, **Current Time(**현재 시간**)** 탭에서 전체 동기화 상태를 볼 수 있습니다. 또는 **Time Synchronization(**시간 동기화**)** 탭에 있는 **NTP Server(NTP** 서버**)** 테이블의 Server Status(서버 상태) 필드를 확인하여 구성된 각 NTP 서버의 동기화 상태를 볼 수 있습니다. 시스템을 특정 NTP 서버와 동기화할 수 없는 경우 Server Status(서버 상태) 옆에 있는 정보 아이콘에 마우스 커서를 대면 자세한 내용을 확인할 수 있습니다.

구성된 날짜 및 시간 보기

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 14 페이지 참고).

단계 **2** 다음 명령을 사용하여 구성된 표준 시간대를 확인합니다.
Firepower-chassis# **showtimezone**

단계 **3** 다음 명령을 사용하여 구성된 날짜 및 시간을 확인합니다.
Firepower-chassis# **showclock**

다음 예에서는 구성된 표준 시간대와 현재 시스템 날짜 및 시간을 표시하는 방법을 보여줍니다.

```
Firepower-chassis# show timezone
Timezone: America/Chicago
Firepower-chassis# show clock
Thu Jun  2 12:40:42 CDT 2016
Firepower-chassis#
```

표준 시간대 설정

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis# **scopesystem**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system # **scopeservices**

단계 **3** 표준 시간대를 설정합니다.
Firepower-chassis /system/services # **settimezone**

이때 사용자의 대륙, 국가 및 표준 시간대 지역에 해당하는 숫자를 입력하라는 프롬프트가 표시됩니다. 각 프롬프트에 적절한 정보를 입력합니다.

위치 정보 지정을 완료하고 나면 올바른 표준 시간대 정보를 설정 중인지 확인하라는 프롬프트가 표시됩니다. 1(예)을 입력하여 확인하거나 2(아니오)를 입력하여 작업을 취소합니다.

단계 **4** 다음 명령을 사용하여 구성된 표준 시간대를 확인합니다.
Firepower-chassis /system/services # **top**
Firepower-chassis# **show timezone**

다음 예에서는 표준 시간대를 태평양 표준 시간대 지역으로 구성하고, 트랜잭션을 커밋한 다음, 구성된 표준 시간대를 표시합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # set timezone
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa            4) Arctic Ocean     7) Australia       10) Pacific Ocean
2) Americas          5) Asia             8) Europe
3) Antarctica        6) Atlantic Ocean   9) Indian Ocean
#? 2
Please select a country.
 1) Anguilla                 28) Haiti
 2) Antigua & Barbuda        29) Honduras
 3) Argentina                30) Jamaica
 4) Aruba                    31) Martinique
 5) Bahamas                  32) Mexico
 6) Barbados                 33) Montserrat
 7) Belize                   34) Nicaragua
 8) Bolivia                  35) Panama
 9) Brazil                   36) Paraguay
10) Canada                   37) Peru
11) Caribbean Netherlands    38) Puerto Rico
12) Cayman Islands           39) St Barthelemy
13) Chile                    40) St Kitts & Nevis
14) Colombia                 41) St Lucia
15) Costa Rica               42) St Maarten (Dutch part)
16) Cuba                     43) St Martin (French part)
17) Curacao                  44) St Pierre & Miquelon
18) Dominica                 45) St Vincent
19) Dominican Republic       46) Suriname
20) Ecuador                  47) Trinidad & Tobago
21) El Salvador              48) Turks & Caicos Is
22) French Guiana            49) United States
23) Greenland                50) Uruguay
24) Grenada                  51) Venezuela
25) Guadeloupe               52) Virgin Islands (UK)
26) Guatemala                53) Virgin Islands (US)
27) Guyana
#? 49
Please select one of the following time zone regions.
 1) Eastern Time
 2) Eastern Time - Michigan - most locations
 3) Eastern Time - Kentucky - Louisville area
 4) Eastern Time - Kentucky - Wayne County
 5) Eastern Time - Indiana - most locations
 6) Eastern Time - Indiana - Daviess, Dubois, Knox & Martin Counties
 7) Eastern Time - Indiana - Pulaski County
 8) Eastern Time - Indiana - Crawford County
 9) Eastern Time - Indiana - Pike County
10) Eastern Time - Indiana - Switzerland County
11) Central Time
12) Central Time - Indiana - Perry County
13) Central Time - Indiana - Starke County
14) Central Time - Michigan - Dickinson, Gogebic, Iron & Menominee Counties
15) Central Time - North Dakota - Oliver County
16) Central Time - North Dakota - Morton County (except Mandan area)
17) Central Time - North Dakota - Mercer County
18) Mountain Time
19) Mountain Time - south Idaho & east Oregon
20) Mountain Standard Time - Arizona (except Navajo)
21) Pacific Time
22) Pacific Standard Time - Annette Island, Alaska
23) Alaska Time
24) Alaska Time - Alaska panhandle
25) Alaska Time - southeast Alaska panhandle
26) Alaska Time - Alaska panhandle neck
27) Alaska Time - west Alaska
28) Aleutian Islands
29) Hawaii
#? 21
```

```
The following information has been given:

        United States
        Pacific Time

Therefore timezone 'America/Los_Angeles' will be set.
Local time is now:      Wed Jun 24 07:39:25 PDT 2015.
Universal Time is now:  Wed Jun 24 14:39:25 UTC 2015.
Is the above information OK?
1) Yes
2) No
#? 1
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services # top
Firepower-chassis# show timezone
Timezone: America/Los_Angeles (Pacific Time)
Firepower-chassis#
```

NTP를 사용하여 날짜 및 시간 설정

NTP는 네트워크 시스템 간에 정확하게 동기화된 시간을 제공하는 계층적 서버 시스템을 구현하는 데 사용합니다. 정밀한 타임 스탬프가 포함된 CRL 검증과 같이 시간에 민감한 작업에는 이러한 정확성이 필요합니다. 최대 4개의 NTP 서버를 구성할 수 있습니다.

참고 FXOS 2.2(2) 이상 버전에서는 NTP 버전 3을 사용합니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis# **scopesystem**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system # **scopeservices**

단계 **3** 지정된 호스트 이름, IPv4 또는 IPv6 주소가 있는 NTP 서버를 사용하도록 시스템을 구성합니다.
Firepower-chassis /system/services # **createntp-server** *{hostname | ip-addr | ip6-addr}*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /system/services # **commit-buffer**

단계 **5** 다음 명령을 사용하여 구성된 모든 NTP 서버의 동기화 상태를 확인합니다.
Firepower-chassis /system/services # **show ntp-server**

단계 **6** 다음 명령을 사용하여 특정 NTP 서버의 동기화 상태를 확인합니다.
Firepower-chassis /system/services # **scopentp-server** *{hostname | ip-addr | ip6-addr}*
Firepower-chassis /system/services/ntp-server # **show detail**

다음 예에서는 IP 주소 192.168.200.101을 사용하는 NTP 서버를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create ntp-server 192.168.200.101
```

```
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

다음 예에서는 IPv6 주소 4001::6을 사용하는 NTP 서버를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create ntp-server 4001::6
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

NTP 서버 삭제

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis# **scopesystem**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system # **scopeservices**

단계 **3** 지정된 호스트 이름, IPv4 또는 IPv6 주소가 있는 NTP 서버를 삭제합니다.
Firepower-chassis /system/services # **deletentp-server** *{hostname | ip-addr | ip6-addr}*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /system/services # **commit-buffer**

다음 예에서는 IP 주소 192.168.200.101을 사용하는 NTP 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete ntp-server 192.168.200.101
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

다음 예에서는 IPv6 주소 4001::6을 사용하는 NTP 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete ntp-server 4001::6
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

날짜 및 시간 수동 설정

이 섹션에서는 Firepower 섀시에 날짜 및 시간을 수동으로 설정하는 방법을 설명합니다. 시스템 클록 수정사항은 즉시 적용됩니다.

참고 시스템 클록이 현재 NTP 서버와 동기화되고 있는 경우, 날짜 및 시간을 수동으로 설정할 수 없습니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis# **scopesystem**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system # **scopeservices**

단계 **3** 시스템 클록을 구성합니다.
Firepower-chassis /system/services # **set clock** *month day year hour min sec*

월의 경우, 월의 첫 세 자릿수를 사용합니다. 시간은 24시간 형식으로 입력해야 합니다(예를 들어, 오후 7시는 19시로 입력).

시스템 클록 수정사항은 즉시 적용됩니다. 버퍼를 커밋할 필요가 없습니다.

다음 예에서는 시스템 클록을 구성합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # set clock jun 24 2015 15 27 00
Firepower-chassis /system/services #
```

SSH 구성

다음 절차에서는 Firepower 섀시에 대한 SSH 액세스를 활성화하거나 비활성화하는 방법과 FXOS 섀시를 SSH 클라이언트로 활성화하는 방법을 설명합니다. SSH는 기본적으로 활성화되어 있습니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis #**scope system**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system #**scope services**

단계 **3** Firepower 섀시에 대한 SSH 액세스를 구성하려면 다음 중 하나를 수행합니다.

- Firepower 섀시에 대한 SSH 액세스를 허용하려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **enable ssh-server**

- Firepower 섀시에 대한 SSH 액세스를 허용하지 않으려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **disable ssh-server**

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower /system/services # **commit-buffer**

다음 예에서는 Firepower 섀시에 대한 SSH 액세스를 활성화하고 트랜잭션을 커밋합니다.

```
Firepower# scope system
Firepower /system # scope services
Firepower /system/services # enable ssh-server
Firepower /system/services* # commit-buffer
Firepower /system/services #
```

텔넷 구성

다음 절차에서는 Firepower 섀시에 대한 텔넷 액세스를 활성화하거나 비활성화하는 방법을 설명합니다. 텔넷은 기본적으로 비활성화되어 있습니다.

참고 텔넷 컨피그레이션은 현재 CLI를 사용하는 경우에만 사용할 수 있습니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis #**scope system**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system #**scope services**

단계 **3** Firepower 섀시에 대한 텔넷 액세스를 구성하려면 다음 중 하나를 수행합니다.

- Firepower 섀시에 대한 텔넷 액세스를 허용하려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **enable telnet-server**

- Firepower 섀시에 대한 텔넷 액세스를 허용하지 않으려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **disable telnet-server**

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower /system/services # **commit-buffer**

다음 예에서는 텔넷을 활성화하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /services # enable telnet-server
Firepower-chassis /services* # commit-buffer
Firepower-chassis /services #
```

SNMP 구성

이 섹션에서는 Firepower 섀시에 SNMP(Simple Network Management Protocol)를 구성하는 방법을 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

SNMP 정보

SNMP(Simple Network Management Protocol)는 애플리케이션 레이어 프로토콜로서 SNMP 관리자와 에이전트 간 통신을 위한 메시지 형식을 제공합니다. SNMP는 네트워크의 디바이스를 모니터링하고 관리할 수 있도록 표준화된 프레임워크 및 공용어를 제공합니다.

SNMP 프레임워크는 세 부분으로 구성됩니다.

- SNMP 관리자 — SNMP를 사용하는 네트워크 디바이스의 활동을 제어하고 모니터링하는 데 사용되는 시스템입니다.
- SNMP 에이전트 — Firepower 섀시에 대한 데이터를 유지하고 필요에 따라 데이터를 SNMP 관리자에게 보고하는 Firepower 섀시 내부의 소프트웨어 구성 요소입니다. Firepower 섀시에는 에이전트 및 MIB 모음이 포함되어 있습니다. SNMP 에이전트를 활성화하고 관리자와 에이전트 간의 관계를 생성하려면 Firepower Chassis Manager 또는 FXOS CLI에서 SNMP를 활성화하고 구성합니다.
- MIB(managed information base) — SNMP 에이전트에 있는 관리 객체의 모음입니다.

Firepower 섀시는 SNMPv1, SNMPv2c 및 SNMPv3를 지원합니다. SNMPv1 및 SNMPv2c는 모두 커뮤니티 기반 보안 유형을 사용합니다. SNMP는 다음과 같이 정의됩니다.

- RFC 3410(http://tools.ietf.org/html/rfc3410)
- RFC 3411(http://tools.ietf.org/html/rfc3411)
- RFC 3412(http://tools.ietf.org/html/rfc3412)
- RFC 3413(http://tools.ietf.org/html/rfc3413)
- RFC 3414(http://tools.ietf.org/html/rfc3414)
- RFC 3415(http://tools.ietf.org/html/rfc3415)
- RFC 3416(http://tools.ietf.org/html/rfc3416)
- RFC 3417(http://tools.ietf.org/html/rfc3417)
- RFC 3418(http://tools.ietf.org/html/rfc3418)
- RFC 3584(http://tools.ietf.org/html/rfc3584)

SNMP 알림

SNMP의 핵심 기능 중 하나는 SNMP 에이전트의 알림을 생성하는 것입니다. 이러한 알림에는 SNMP 관리자가 요청을 전송하지 않아도 됩니다. 알림은 잘못된 사용자 인증, 재시작, 연결 종료, 네이버 라우터와의 연결 끊김, 기타 중대한 이벤트를 나타낼 수 있습니다.

Firepower 섀시는 SNMP 알림을 트랩 또는 정보로 생성합니다. SNMP 관리자가 트랩을 수신할 때 승인을 전송하지 않으며 Firepower 섀시에서는 트랩 수신 여부를 확인할 수 없으므로 트랩은 정보보다 신뢰성이 떨어집니다. 정보 요청을 수신한 SNMP 관리자는 SNMP 응답 PDU(Protocol Data Unit)로 메시지를 승인합니다. Firepower 섀시에서 PDU를 수신하지 않을 경우 정보 요청을 다시 보낼 수 있습니다.

SNMP 보안 레벨 및 권한

SNMPv1, SNMPv2c, SNMPv3은 각각 서로 다른 보안 모델을 나타냅니다. 보안 모델 및 선택된 보안 레벨의 조합을 통해 SNMP 메시지 처리 시 적용할 보안 메커니즘이 결정됩니다.

보안 레벨은 SNMP 트랩과 연결된 메시지를 보는 데 필요한 권한을 결정합니다. 권한 수준은 메시지가 공개되지 않도록 보호해야 하는지 또는 인증되어야 하는지를 결정합니다. 지원되는 보안 레벨은 어떤 보안 모델이 구현되었는지에 따라 달라집니다. SNMP 보안 레벨은 다음 권한을 하나 이상 지원합니다.

- noAuthNoPriv — 인증도 암호화도 없음
- authNoPriv — 인증은 있지만 암호화 없음
- authPriv — 인증 및 암호화 있음

SNMPv3는 보안 모델 및 보안 레벨을 모두 제공합니다. 보안 모델은 사용자 및 사용자가 속한 역할에 대해 설정되는 인증 전략입니다. 보안 레벨은 보안 모델 내에서 허용된 보안 레벨입니다. 보안 모델과 보안 레벨의 조합을 통해 SNMP 패킷 처리 시 적용할 보안 메커니즘이 결정됩니다.

지원되는 SNMP 보안 모델 및 레벨의 조합

다음 표에서는 보안 모델 및 레벨의 조합에 대해 설명합니다.

표 8: SNMP 보안 모델 및 레벨

모델	레벨	인증	암호화	결과
v1	noAuthNoPriv	커뮤니티 문자열	아니요	인증에 커뮤니티 문자열 일치를 사용합니다.

모델	레벨	인증	암호화	결과
v2c	noAuthNoPriv	커뮤니티 문자열	아니요	인증에 커뮤니티 문자열 일치를 사용합니다.
v3	noAuthNoPriv	사용자 이름	아니요	인증에 사용자 이름 일치를 사용합니다.
v3	authNoPriv	HMAC-SHA	아니요	HMAC SHA(Secure Hash Algorithm)를 기반으로 인증을 제공합니다.
v3	authPriv	HMAC-SHA	DES	HMAC-SHA 알고리즘을 기반으로 인증을 제공합니다. DES(Data Encryption Standard) 56비트 암호화 및 CBC(Cipher Block Chaining) DES(DES-56) 표준 기반의 인증을 제공합니다.

SNMPv3 보안 기능

SNMPv3에서는 네트워크를 통한 인증 프레임과 암호화 프레임의 조합을 통해 디바이스에 대한 보안 액세스를 제공합니다. SNMPv3에서는 구성된 사용자에 의한 관리 작업만 승인하고 SNMP 메시지를 암호화합니다. SNMPv3 USM(User-Based Security Model)은 SNMP 메시지 레벨 보안을 가리키며 다음 서비스를 제공합니다.

- 메시지 무결성 — 메시지가 무단으로 변경 또는 손상되지 않았으며 데이터 시퀀스가 악의 없이 수행된 변경보다 적게 변경되었음을 보장합니다.
- 메시지 출처 인증 — 수신한 데이터의 출처를 대신하는 사용자의 클레임된 ID가 확인되었음을 보장합니다.
- 메시지 기밀 보호 및 암호화 — 미승인 개인, 엔터티 또는 프로세스에 정보가 제공되거나 공개되지 않도록 합니다.

SNMP 지원

Firepower 섀시는 SNMP에 다음을 지원합니다.

MIB 지원

Firepower 섀시는 MIB에 대해 읽기 전용 액세스를 지원합니다.

SNMPv3 사용자를 위한 인증 프로토콜

Firepower 섀시는 SNMPv3 사용자를 위해 HMAC-SHA-96(SHA) 인증 프로토콜을 지원합니다.

SNMPv3 사용자를 위한 AES 개인 프로토콜

Firepower 섀시는 SNMPv3 메시지 암호화를 위한 개인 프로토콜 중 하나로 AES(Advanced Encryption Standard)를 사용하며 RFC 3826을 준수합니다.

개인 비밀번호(priv) 옵션에서는 SNMP 보안 암호화를 위해 DES 또는 128비트 AES 암호화를 선택할 수 있습니다. AES-128 컨피그레이션을 활성화하는 경우 SNMPv3 사용자를 위한 개인 비밀번호가 있으면 Firepower 섀시에서는 개인 비밀번호를 사용하여 128비트 AES 키를 생성합니다. AES 개인 비밀번호는 최소 8자입니다. 비밀번호가 일반 텍스트로 지정된 경우 최대 64자로 지정할 수 있습니다.

SNMP 활성화 및 SNMP 속성 구성

절차

단계 **1** 모니터링 모드를 시작합니다.
Firepower-chassis# **scope monitoring**

단계 **2** SNMP를 활성화합니다.
Firepower-chassis /monitoring # **enable snmp**

단계 **3** SNMP 커뮤니티 모드를 시작합니다.
Firepower-chassis /monitoring # **set snmp community**

set snmp community 명령을 입력한 후 SNMP 커뮤니티를 입력하라는 프롬프트가 표시됩니다.

단계 **4** SNMP 커뮤니티를 지정합니다. 커뮤니티 이름을 비밀번호로 사용합니다. 커뮤니티 이름은 최대 32자의 영숫자 문자열이 될 수 있습니다.
Firepower-chassis /monitoring # **Enter a snmp community:** *community-name*

단계 **5** SNMP를 책임지는 시스템 담당자를 지정합니다. 시스템 연락처 이름은 이메일 주소 또는 이름과 전화번호 등 최대 255자의 영숫자 문자열이 될 수 있습니다.
Firepower-chassis /monitoring # **set snmp syscontact** *system-contact-name*

단계 **6** SNMP 에이전트(서버)가 실행되는 호스트의 위치를 지정합니다. 시스템 위치 이름은 최대 512자의 영숫자 문자열이 될 수 있습니다.
Firepower-chassis /monitoring # **set snmp syslocation** *system-location-name*

단계 **7** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /monitoring # **commit-buffer**

다음 예에서는 SNMP를 활성화하고, SnmpCommSystem2라는 이름의 SNMP 커뮤니티를 구성한 다음, contactperson이라는 이름의 시스템 담당자를 구성하고, systemlocation이라는 이름의 연락처 위치를 구성한 후, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # set snmp community
Enter a snmp community: SnmpCommSystem2
Firepower-chassis /monitoring* # set snmp syscontact contactperson1
Firepower-chassis /monitoring* # set snmp syslocation systemlocation
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

다음에 할 작업

SNMP 트랩 및 사용자를 생성합니다.

SNMP 트랩 생성

절차

단계 **1** 모니터링 모드를 시작합니다.
Firepower-chassis# **scope monitoring**

단계 **2** SNMP를 활성화합니다.
Firepower-chassis /monitoring # **enable snmp**

단계 **3** 지정된 호스트 이름, IPv4 주소 또는 IPv6 주소로 SNMP 트랩을 생성합니다.
Firepower-chassis /monitoring # **create snmp-trap** {*hostname* | *ip-addr* | *ip6-addr*}

단계 **4** SNMP 트랩에 사용할 SNMP 커뮤니티 이름을 지정합니다.
Firepower-chassis /monitoring/snmp-trap # **set community** *community-name*

단계 **5** SNMP 트랩에 사용할 포트를 지정합니다.
Firepower-chassis /monitoring/snmp-trap # **set port** *port-num*

단계 **6** 트랩에 사용되는 SNMP 버전 및 모델을 지정합니다.
Firepower-chassis /monitoring/snmp-trap # **set version** {**v1** | **v2c** | **v3**}

단계 **7** (선택 사항) 전송할 트랩 유형을 지정합니다.
Firepower-chassis /monitoring/snmp-trap # **set notificationtype** {**traps** | **informs**}

다음을 선택할 수 있습니다.

- 버전으로 v2c 또는 v3를 선택할 경우 **traps**(트랩)
- 버전으로 v2c를 선택할 경우 **informs**(정보)

참고 정보 알림은 v2c 버전을 선택한 경우에만 전송될 수 있습니다.

단계 **8** (선택 사항) v3 버전을 선택하는 경우 트랩과 연관된 권한을 지정합니다.
Firepower-chassis /monitoring/snmp-trap # **set v3privilege** {**auth** | **noauth** | **priv**}

다음을 선택할 수 있습니다.

- **auth** — 인증은 있지만 암호화 없음
- **noauth** — 인증도 암호화도 없음
- **priv** — 인증 및 암호화

단계 **9** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /monitoring/snmp-trap # **commit-buffer**

다음 예에서는 SNMP를 활성화하고 IPv4 주소를 사용하여 SNMP 트랩을 생성하고 트랩이 port 2에서 SnmpCommSystem2 커뮤니티를 사용하도록 지정하고 버전을 v3로 설정하고 알림 유형을 트랩으로 설정하고 v3 권한을 priv로 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # create snmp-trap 192.168.100.112
Firepower-chassis /monitoring/snmp-trap* # set community SnmpCommSystem2
Firepower-chassis /monitoring/snmp-trap* # set port 2
Firepower-chassis /monitoring/snmp-trap* # set version v3
Firepower-chassis /monitoring/snmp-trap* # set notificationtype traps
Firepower-chassis /monitoring/snmp-trap* # set v3privilege priv
Firepower-chassis /monitoring/snmp-trap* # commit-buffer
Firepower-chassis /monitoring/snmp-trap #
```

다음 예에서는 SNMP를 활성화하고 IPv6 주소를 사용하여 SNMP 트랩을 생성하고 트랩이 port 2에서 SnmpCommSystem3 커뮤니티를 사용하도록 지정하고 버전을 v3로 설정하고 알림 유형을 트랩으로 설정하고 v3 권한을 priv로 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # create snmp-trap 2001::1
Firepower-chassis /monitoring/snmp-trap* # set community SnmpCommSystem3
Firepower-chassis /monitoring/snmp-trap* # set port 2
Firepower-chassis /monitoring/snmp-trap* # set version v3
Firepower-chassis /monitoring/snmp-trap* # set notificationtype traps
Firepower-chassis /monitoring/snmp-trap* # set v3privilege priv
Firepower-chassis /monitoring/snmp-trap* # commit-buffer
Firepower-chassis /monitoring/snmp-trap #
```

SNMP 트랩 삭제

절차

단계 **1** 모니터링 모드를 시작합니다.

Firepower-chassis# **scope monitoring**

단계 **2** 지정된 호스트 이름 또는 IP 주소가 있는 SNMP 트랩을 삭제합니다.
Firepower-chassis /monitoring # **delete snmp-trap** {*hostname* | *ip-addr*}

단계 **3** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /monitoring # **commit-buffer**

다음 예에서는 IP 주소 192.168.100.112의 SNMP 트랩을 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # delete snmp-trap 192.168.100.112
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

SNMPv3 사용자 생성

절차

단계 **1** 모니터링 모드를 시작합니다.
Firepower-chassis# **scope monitoring**

단계 **2** SNMP를 활성화합니다.
Firepower-chassis /monitoring # **enable snmp**

단계 **3** 지정된 SNMPv3 사용자를 생성합니다.
Firepower-chassis /monitoring # **create snmp-user** *user-name*

create snmp-user 명령을 입력하고 나면 비밀번호를 입력하라는 프롬프트가 표시됩니다.

단계 **4** AES-128 암호화 사용을 활성화 또는 비활성화합니다.
Firepower-chassis /monitoring/snmp-user # **set aes-128** {**no** | **yes**}

기본적으로 AES-128 암호화는 비활성화되어 있습니다.

단계 **5** 사용자 개인 비밀번호를 지정합니다.
Firepower-chassis /monitoring/snmp-user # **set priv-password**

set priv-password 명령을 입력하고 나면 개인 비밀번호를 입력하고 확인하라는 프롬프트가 표시됩니다.

단계 **6** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /monitoring/snmp-user # **commit-buffer**

다음 예에서는 SNMP를 활성화하고, snmp-user14라는 이름의 SNMPv3 사용자를 생성한 다음, AES-128 암호화를 활성화하고, 비밀번호 및 개인 비밀번호를 설정한 후, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # create snmp-user snmp-user14
```

```
Password:
Firepower-chassis /monitoring/snmp-user* # set aes-128 yes
Firepower-chassis /monitoring/snmp-user* # set priv-password
Enter a password:
Confirm the password:
Firepower-chassis /monitoring/snmp-user* # commit-buffer
Firepower-chassis /monitoring/snmp-user #
```

SNMPv3 사용자 삭제

절차

단계 **1** 모니터링 모드를 시작합니다.
Firepower-chassis# **scope monitoring**

단계 **2** 지정된 SNMPv3 사용자를 삭제합니다.
Firepower-chassis /monitoring # **delete snmp-user** *user-name*

단계 **3** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /monitoring # **commit-buffer**

다음 예에서는 snmp-user14라는 이름의 SNMPv3 사용자를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # delete snmp-user snmp-user14
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

HTTPS 구성

이 섹션에서는 Firepower 4100/9300 섀시에 HTTPS를 구성하는 방법을 설명합니다.

참고 Firepower Chassis Manager 또는 FXOS CLI를 사용하여 HTTPS 포트를 변경할 수 있습니다. 다른 모든 HTTPS 컨피그레이션은 FXOS CLI를 사용하는 경우에만 수행할 수 있습니다.

인증서, 키 링 및 트러스트 포인트

HTTPS에서는 PKI(Public Key Infrastructure)의 구성 요소를 사용하여 두 디바이스, 이를테면 클라이언트의 브라우저와 Firepower 4100/9300 섀시 간의 보안 통신을 설정합니다.

암호화 키 및 키 링

각 PKI 디바이스는 비대칭 RSA(Rivest-Shamir-Adleman) 암호화 키의 쌍을 보유합니다. 개인 키와 공개 키로 구성된 이 쌍은 내부 키 링에 저장됩니다. 두 키 중 하나로 암호화한 메시지는 나머지 키로 해독할 수 있습니다. 암호화된 메시지를 보낼 때 발신자는 수신자의 공개 키로 메시지를 암호화하며 수

신자는 자신의 개인 키로 그 메시지를 해독합니다. 발신자는 알려진 메시지를 자신의 개인 키로 암호화('서명'이라고도 함)하여 공개 키에 대한 소유권을 입증할 수도 있습니다. 수신자가 해당 공개 키를 사용하여 성공적으로 메시지를 해독할 수 있다면 발신자가 해당 개인 키를 소유하고 있음이 입증됩니다. 암호화 키의 길이는 다양하지만, 일반적으로 512~2048비트입니다. 일반적으로 키는 길수록 더 안전합니다. FXOS에서는 초기 2048비트 키 쌍으로 기본 키 링을 제공하며, 사용자가 추가 키 링을 생성할 수 있습니다.

클러스터 이름이 변경되거나 인증서가 만료되는 경우 기본 키 링 인증서를 수동으로 재생성해야 합니다.

인증서

안전한 통신을 위해 두 디바이스는 먼저 디지털 인증서를 교환합니다. 인증서는 디바이스 ID에 대해 서명된 정보와 함께 디바이스 공개 키를 포함하는 파일입니다. 단순히 암호화된 통신을 지원하기 위해 디바이스에서 자신의 키 쌍 및 자체 서명된 인증서를 생성할 수 있습니다. 원격 사용자가 자체 서명 인증서가 있는 디바이스에 연결하는 경우 사용자는 디바이스의 ID를 쉽게 확인할 방법이 없으며 사용자의 브라우저는 초기에 인증 경고를 표시합니다. 기본적으로 FXOS에는 기본 키 링의 공개 키를 포함하는 자체 서명 인증서가 내장되어 있습니다.

트러스트 포인트

신뢰할 수 있는 출처 또는 트러스트 포인트로부터 디바이스의 ID를 확인하는 서드파티 인증서를 얻어 설치하면 FXOS에 대해 더 강력한 인증을 제공할 수 있습니다. 서드파티 인증서는 해당 인증서를 발급하는 트러스트 포인트에서 서명되며, 루트 CA(인증 기관), 중간 CA 또는 루트 CA로 연결되는 트러스트 체인의 일부인 Trust Anchor가 될 수 있습니다. 새 인증서를 얻으려면 FXOS를 통해 인증서 요청을 생성하고 트러스트 포인트에 요청을 제출해야 합니다.

중요 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.

키 링 생성

FXOS는 기본 키 링을 포함하여 최대 8개의 키 링을 지원합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 키 링을 생성하고 이름을 지정합니다.
Firepower-chassis # **createkeyring** *keyring-name*

단계 **3** SSL 키 길이(비트)를 설정합니다.
Firepower-chassis # **setmodulus** {**mod1024** | **mod1536** | **mod2048** | **mod512**}

단계 **4** 트랜잭션을 커밋합니다.

Firepower-chassis # **commit-buffer**

다음 예에서는 키 크기가 1024비트인 키 링을 생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create keyring kr220
Firepower-chassis /security/keyring* # set modulus mod1024
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 할 작업

이 키 링에 대한 인증서 요청을 생성합니다.

기본 키 링 재생성

클러스터 이름이 변경되거나 인증서가 만료되는 경우 기본 키 링 인증서를 수동으로 재생성해야 합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 기본 키 링의 키 링 보안 모드를 시작합니다.
Firepower-chassis /security # **scopekeyring default**

단계 **3** 기본 키 링을 재생성합니다.
Firepower-chassis /security/keyring # **setregenerate yes**

단계 **4** 트랜잭션을 커밋합니다.
Firepower-chassis # **commit-buffer**

다음 예에서는 기본 키 링을 재생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring default
Firepower-chassis /security/keyring* # set regenerate yes
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

키 링에 대한 인증서 요청 생성

기본 옵션을 사용하여 키 링에 대한 인증서 요청 생성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 키 링의 컨피그레이션 모드를 시작합니다.
Firepower-chassis /security # **scope keyring** *keyring-name*

단계 **3** 지정된 IPv4 또는 IPv6 주소 또는 Fabric Interconnect의 이름을 사용하여 인증서 요청을 생성합니다. 인증서 요청에 대한 비밀번호를 입력하라는 프롬프트가 표시됩니다.
Firepower-chassis /security/keyring # **create certreq** {**ip** [*ipv4-addr* | *ipv6-v6*] |**subject-name** *name*}

단계 **4** 트랜잭션을 커밋합니다.
Firepower-chassis /security/keyring/certreq # **commit-buffer**

단계 **5** 인증서 요청을 표시합니다. 이 요청은 복사하여 Trust Anchor 또는 인증 기관(CA)에 보낼 수 있습니다.
Firepower-chassis /security/keyring # **show certreq**

다음 예에서는 기본 옵션을 사용하여 키 링에 대한 IPv4 주소로 인증서 요청을 생성하고 표시합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq ip 192.168.200.123 subject-name sjc04
Certificate request password:
Confirm certificate request password:
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name:
Certificate request country name:
State, province or county (full name):
Locality (eg, city):
Organization name (eg, company):
Organization Unit name (eg, section):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring #
```

다음에 할 작업

- BEGIN(시작) 및 END(끝) 라인을 포함하여 인증서 요청의 텍스트를 복사하여 파일에 저장합니다. 키 링에 대한 인증서를 얻으려면 파일을 인증서 요청과 함께 Trust Anchor 또는 인증 기관(CA)에 전송합니다.
- 트러스트 포인트를 생성하고 Trust Anchor로부터 받은 신뢰 인증서에 대한 인증서 체인을 설정합니다.

고급 옵션을 사용하여 키 링에 대한 인증서 요청 생성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 키 링의 컨피그레이션 모드를 시작합니다.
Firepower-chassis /security # **scope keyring** *keyring-name*

단계 **3** 인증서 요청을 생성합니다.
Firepower-chassis /security/keyring # **createcertreq**

단계 **4** 회사가 위치한 국가의 국가 코드를 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set country** *country name*

단계 **5** 요청과 연결된 DNS(Domain Name Server) 주소를 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set dns** *DNS Name*

단계 **6** 인증서 요청과 연결된 이메일 주소를 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set e-mail** *E-mail name*

단계 **7** Firepower 4100/9300 섀시의 IP 주소를 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set ip** {*certificate request ip-address*|*certificate request ip6-address* }

단계 **8** 인증서를 요청하는 회사의 본사가 위치한 도시 또는 지역을 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set locality** *locality name (eg, city)*

단계 **9** 인증서를 요청하는 조직을 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set org-name** *organization name*

단계 **10** 조직 단위를 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set org-unit-name** *organizational unit name*

단계 **11** 인증서 요청에 대한 비밀번호를 지정합니다(선택 사항).
Firepower-chassis /security/keyring/certreq* # **set password** *certificate request password*

단계 **12** 인증서를 요청하는 회사의 본사가 위치한 시/도를 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set state** *state, province or county*

단계 **13** Firepower 4100/9300 섀시의 FQDN(Fully Qualified Domain Name)을 지정합니다.
Firepower-chassis /security/keyring/certreq* # **set subject-name** *certificate request name*

단계 **14** 트랜잭션을 커밋합니다.
Firepower-chassis /security/keyring/certreq # **commit-buffer**

단계 **15** 인증서 요청을 표시합니다. 이 요청은 복사하여 Trust Anchor 또는 인증 기관(CA)에 보낼 수 있습니다.
Firepower-chassis /security/keyring # **show certreq**

다음 예에서는 고급 옵션을 사용하여 키 링에 대한 IPv4 주소로 인증서 요청을 생성하고 표시합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq
Firepower-chassis /security/keyring/certreq* # set ip 192.168.200.123
Firepower-chassis /security/keyring/certreq* #  set subject-name sjc04
Firepower-chassis /security/keyring/certreq* # set country US
Firepower-chassis /security/keyring/certreq* # set dns bg1-samc-15A
Firepower-chassis /security/keyring/certreq* # set email test@cisco.com
Firepower-chassis /security/keyring/certreq* # set locality new york city
Firepower-chassis /security/keyring/certreq* # set org-name "Cisco Systems"
Firepower-chassis /security/keyring/certreq* # set org-unit-name Testing
Firepower-chassis /security/keyring/certreq* # set state new york
Firepower-chassis /security/keyring/certreq* # commit-buffer
Firepower-chassis /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name: test@cisco.com
Certificate request country name: US
State, province or county (full name): New York
Locality name (eg, city): new york city
Organization name (eg, company): Cisco
Organization Unit name (eg, section): Testing
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring/certreq #
```

다음에 할 작업

- BEGIN(시작) 및 END(끝) 라인을 포함하여 인증서 요청의 텍스트를 복사하여 파일에 저장합니다. 키 링에 대한 인증서를 얻으려면 파일을 인증서 요청과 함께 Trust Anchor 또는 인증 기관(CA)에 전송합니다.
- 트러스트 포인트를 생성하고 Trust Anchor로부터 받은 신뢰 인증서에 대한 인증서 체인을 설정합니다.

트러스트 포인트 생성

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 트러스트 포인트를 생성합니다.
Firepower-chassis /security # **createtrustpoint** *name*

단계 **3** 이 트러스트 포인트에 대한 인증서 정보를 지정합니다.
Firepower-chassis /security/trustpoint # **setcertchain** [*certchain*]

명령에 인증서 정보를 지정하지 않은 경우, 루트 CA(인증 기관)에 인증 경로를 정의하는 트러스트 포인트 목록 또는 인증서를 입력하라는 프롬프트가 표시됩니다. 해당 정보를 입력한 후 다음 행에 ENDOFBUF를 입력하여 완료합니다.

중요 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.

단계 **4** 트랜잭션을 커밋합니다.
Firepower-chassis /security/trustpoint # **commit-buffer**

다음 예에서는 트러스트 포인트를 생성하고 트러스트 포인트에 대한 인증서를 제공합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create trustpoint tPoint10
Firepower-chassis /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMDCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKOND1
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> jtcEMyZ+f7+3yh421ido3nO4MIGeBgNVHSMEgZYwgZOAFLlNjtcEMyZ+f7+3yh42
> 1ido3nO4oXikdjB0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAcT
> C1NhbnRhIENsYXJhMRswGQYDVQQKExJOdW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0VuZ2luZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAQAwDAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGQqXc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8copP1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIZJwK5+N3x/kfa2EHU6id1avt
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/trustpoint* # commit-buffer
Firepower-chassis /security/trustpoint #
```

다음에 할 작업

Trust anchor 또는 인증 기관에서 인증서를 수신하면 키 링으로 가져옵니다.

키 링에 인증서 가져오기

시작하기 전에

- 키 링 인증서에 대한 인증서 체인을 포함하는 트러스트 포인트를 구성합니다.
- Trust Anchor 또는 인증 기관(CA)에서 키 링 인증서를 얻습니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 인증서를 수신할 키 링에 대한 컨피그레이션 모드로 들어갑니다.
Firepower-chassis /security # **scopekeyring** *keyring-name*

단계 **3** 키 링 인증서를 수신한 Trust anchor 또는 인증 기관(CA)에 대한 트러스트 포인트를 지정합니다.
Firepower-chassis /security/keyring # **settrustpoint** *name*

단계 **4** 키 링 인증서를 입력하고 업로드하기 위해 대화 상자를 실행합니다.
Firepower-chassis /security/keyring # **setcert**

프롬프트에서 Trust Anchor 또는 인증 기관(CA)에서 받은 인증서 텍스트를 붙여넣습니다. 인증서의 다음 행에 ENDOFBUF를 입력하여 인증서 입력을 완료합니다.

중요 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.

단계 **5** 트랜잭션을 커밋합니다.
Firepower-chassis /security/keyring # **commit-buffer**

다음 예에서는 트러스트 포인트를 지정하고 키 링에 인증서를 가져옵니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # set trustpoint tPoint10
Firepower-chassis /security/keyring* # set cert
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
> -----BEGIN CERTIFICATE-----
> MIIB/zCCAWgCAQAwgZkxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTEVMBMGA1UE
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKONDl
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L43l7IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 할 작업

HTTPS 서비스를 키 링으로 구성합니다.

HTTPS 구성

주의 HTTPS에서 사용하는 포트 및 키 링 변경을 포함하여 HTTPS 컨피그레이션을 완료한 후 트랜잭션을 저장하거나 커밋하자마자 모든 현재 HTTP 및 HTTPS 세션이 종료됩니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis# **scope system**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system # **scope services**

단계 **3** HTTPS 서비스를 활성화합니다.
Firepower-chassis /system/services # **enable https**

단계 **4** (선택 사항) HTTPS 연결에 사용할 포트를 지정합니다.
Firepower-chassis /system/services # **set https port** *port-num*

단계 **5** (선택 사항) HTTPS용으로 생성한 키 링의 이름을 지정합니다.
Firepower-chassis /system/services # **set https keyring** *keyring-name*

단계 **6** (선택 사항) 도메인에서 사용하는 Cipher Suite 보안 레벨을 지정합니다.
Firepower-chassis /system/services # **set https cipher-suite-mode** *cipher-suite-mode*

*cipher-suite-mode*는 다음 키워드 중 하나가 될 수 있습니다.

- **high-strength**
- **medium-strength**
- **low-strength**
- **custom**— 사용자 정의 Cipher Suite 사양 문자열을 지정할 수 있습니다.

단계 **7** (선택 사항) **cipher-suite-mode**가 **custom**으로 설정된 경우, 도메인에 대해 Cipher Suite 보안의 맞춤형 레벨을 지정합니다.
Firepower-chassis /system/services # **set https cipher-suite** *cipher-suite-spec-string*

*cipher-suite-spec-string*은 최대 256자이며 OpenSSL Cipher Suite 사양을 준수해야 합니다. 공백 또는 특수 문자를 사용할 수 없습니다. 단, !(느낌표), +(덧셈 기호), -(하이픈), :(콜론)은 사용할 수 있습니다. 자세한 내용은 http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite를 참조하십시오.

예를 들어 FXOS에서 기본값으로 사용하는 중간 강도 사양 문자열은 다음과 같습니다.
ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL

참고 이 옵션은 **cipher-suite-mode**가 **custom**이 아닌 값으로 설정된 경우 무시됩니다.

단계 **8** (선택 사항) 인증서 해지 목록 확인을 활성화하거나 비활성화합니다.
setrevoke-policy *{ relaxed | strict }*

단계 **9** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /system/services # **commit-buffer**

다음 예에서는 HTTPS를 활성화하고, 포터 번호를 443으로 설정하고, 키 링 이름을 kring7984로 설정하고, Cipher Suite 보안 레벨을 high로 설정하고, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # enable https
Firepower-chassis /system/services* # set https port 443
Warning: When committed, this closes all the web sessions.
Firepower-chassis /system/services* # set https keyring kring7984
Firepower-chassis /system/services* # set https cipher-suite-mode high
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

HTTPS 포트 변경

HTTPS 서비스는 기본적으로 포트 443에서 활성화되어 있습니다. HTTPS는 비활성화할 수 없지만 HTTPS 연결에 사용할 포트는 변경할 수 있습니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis #**scope system**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system #**scope services**

단계 **3** HTTPS 연결에 사용할 포트를 지정합니다.
Firepower-chassis /system/services # **sethttpsport** *port-number*

*port-number*에 1~65535의 정수를 지정합니다. HTTPS는 기본적으로 포트 443에서 활성화되어 있습니다.

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower /system/services # **commit-buffer**

HTTPS 포트를 변경한 후에는 현재의 모든 HTTPS 세션이 종료됩니다. 사용자는 다음과 같이 새 포트를 사용하여 Firepower Chassis Manager에 다시 로그인해야 합니다.

https://<*chassis_mgmt_ip_address*>:<*chassis_mgmt_port*>

이때 <*chassis_mgmt_ip_address*>는 사용자가 초기 컨피그레이션 중에 입력한 Firepower 섀시의 IP 주소 또는 호스트 이름이며 <*chassis_mgmt_port*>는 방금 구성한 HTTPS 포트입니다.

다음 예에서는 HTTPS 포트 번호를 443으로 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # set https port 444
Warning: When committed, this closes all the web sessions.
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

키 링 삭제

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis #**scope security**

단계 **2** 이름이 지정된 키 링을 삭제합니다.
Firepower-chassis /security # **deletekeyring** *name*

단계 **3** 트랜잭션을 커밋합니다.
Firepower-chassis /security # **commit-buffer**

다음 예에서는 키 링을 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete keyring key10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

트러스트 포인트 삭제

시작하기 전에

트러스트 포인트가 키 링에서 사용되지 않음을 확인합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scopesecurity**

단계 **2** 이름이 지정된 트러스트 포인트를 삭제합니다.
Firepower-chassis /security # **deletetrustpoint** *name*

단계 **3** 트랜잭션을 커밋합니다.
Firepower-chassis /security # **commit-buffer**

다음 예에서는 트러스트 포인트를 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete trustpoint tPoint10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

HTTPS 비활성화

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis# **scope system**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system # **scope services**

단계 **3** HTTPS 서비스를 비활성화합니다.
Firepower-chassis /system/services # **disable https**

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /system/services # **commit-buffer**

다음 예에서는 HTTPS를 비활성화하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # disable https
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

AAA 구성

이 섹션에서는 인증, 권한 부여 및 어카운팅에 대해 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

AAA 정보

AAA는 컴퓨터 리소스에 대한 액세스 제어를 위한 서비스의 집합으로, 정책을 시행하고 사용량을 평가하며 서비스에 대한 청구에 필요한 정보를 제공합니다. 이 과정은 효과적인 네트워크 관리 및 보안을 위해 중요한 부분으로 간주됩니다.

인증

인증은 액세스를 부여하기 전에 보통 사용자 이름과 비밀번호를 입력하도록 요구하는 방식으로 효과적인 사용자 확인 방법을 제공합니다. AAA 서버는 사용자의 인증 자격 증명을 데이터베이스에 저

장된 다른 사용자의 자격 증명과 비교합니다. 자격 증명이 일치하면 사용자는 네트워크에 액세스할 수 있습니다. 자격 증명이 일치하지 않으면, 인증에 실패하고 네트워크 액세스가 거부됩니다.

다음 세션을 포함하는 섀시에 대한 관리 연결을 인증하도록 Firepower 4100/9300 섀시를 구성할 수 있습니다.

- HTTPS
- SSH
- 직렬 콘솔

권한 부여

권한 부여는 정책을 시행하는 프로세스로 사용자의 액세스가 허용된 활동, 리소스 또는 서비스 유형을 판단하는 것입니다. 사용자가 인증되면 해당 사용자는 다양한 액세스 또는 활동 유형에 대한 허가를 받을 수 있습니다.

어카운팅

어카운팅은 사용자가 액세스 중 사용하는 리소스를 측정합니다. 여기에는 시스템 사용 시간, 사용자가 세션 중 보내거나 받는 데이터의 양 등이 포함됩니다. 어카운팅은 세션 통계 및 사용량 정보 기록을 통해 이루어지며 이는 권한 부여 제어, 청구, 경향 분석, 리소스 활용도 및 용량 계획 활동에 사용됩니다.

인증, 권한 부여, 어카운팅 간 상호 작용

인증을 단독으로 사용하거나 권한 부여 및 어카운팅과 함께 사용할 수 있습니다. 권한 부여에서는 항상 사용자를 먼저 인증해야 합니다. 어카운팅을 단독으로 사용하거나 인증 및 권한 부여와 함께 사용할 수 있습니다.

AAA 서버

AAA 서버는 액세스 제어를 위해 사용되는 네트워크 서버입니다. 인증은 사용자를 식별합니다. 권한 부여는 인증된 사용자가 액세스할 수 있는 리소스와 서비스를 결정하는 정책을 구현합니다. 어카운팅은 청구 및 분석을 위해 사용되는 시간과 데이터 리소스를 추적합니다.

로컬 데이터베이스 지원

Firepower 섀시에서는 사용자 프로파일로 채울 수 있는 로컬 데이터베이스를 유지 관리합니다. AAA 서버 대신 로컬 데이터베이스를 사용하여 사용자 인증, 권한 부여 및 어카운팅을 제공할 수 있습니다.

LDAP 제공자 구성

LDAP 제공자 속성 구성

이 작업에서 구성하는 속성은 이 유형의 모든 제공자 연결에 대한 기본 설정입니다. 개별 제공자에 이러한 속성의 설정이 포함되어 있는 경우에는 Firepower eXtensible 운영 체제에서 해당 설정을 사용하고 기본 설정을 무시합니다.

Active Directory를 LDAP 서버로 사용하는 경우에는 Active Directory 서버에서 사용자 어카운트를 생성하여 Firepower eXtensible 운영 체제와 바인딩합니다. 이 어카운트에는 만료되지 않는 비밀번호가 제공되어야 합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 LDAP 모드를 시작합니다.
Firepower-chassis /security # **scope ldap**

단계 **3** 지정된 속성을 포함하는 레코드로 데이터베이스 검색을 제한합니다.
Firepower-chassis /security/ldap # **set attribute** *attribute*

단계 **4** 지정된 고유 이름을 포함하는 레코드로 데이터베이스 검색을 제한합니다.
Firepower-chassis /security/ldap # **set basedn** *distinguished-name*

단계 **5** 지정된 필터를 포함하는 레코드로 데이터베이스 검색을 제한합니다.
Firepower-chassis /security/ldap # **set filter** *filter*

단계 **6** 서버가 다운되었다고 인지할 때까지 시스템이 LDAP 서버의 응답을 대기해야 하는 시간 간격을 설정합니다.
Firepower-chassis /security/ldap # **set timeout** *seconds*

단계 **7** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/ldap # **commit-buffer**

다음 예에서는 LDAP 속성을 CiscoAvPair로, 기본 고유 이름을 "DC=cisco-firepower-aaa3,DC=qalab,DC=com"으로, 필터를 sAMAccountName=$userid로, 시간 초과 간격을 5초로 각각 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap # set attribute CiscoAvPair
Firepower-chassis /security/ldap* # set basedn "DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap* # set filter sAMAccountName=$userid
Firepower-chassis /security/ldap* # set timeout 5
Firepower-chassis /security/ldap* # commit-buffer
Firepower-chassis /security/ldap #
```

참고 사용자 로그인은 LDAP 사용자의 userdn이 255자를 초과하는 경우 실패합니다.

다음에 할 작업

LDAP 제공자를 생성합니다.

LDAP 제공자 생성

Firepower eXtensible 운영 체제에서는 최대 16개의 LDAP 제공자를 지원합니다.

시작하기 전에

Active Directory를 LDAP 서버로 사용하는 경우에는 Active Directory 서버에서 사용자 어카운트를 생성하여 Firepower eXtensible 운영 체제와 바인딩합니다. 이 어카운트에는 만료되지 않는 비밀번호가 제공되어야 합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 LDAP 모드를 시작합니다.
Firepower-chassis /security # **scope ldap**

단계 **3** LDAP 서버 인스턴스를 생성하고 보안 LDAP 서버 모드를 시작합니다.
Firepower-chassis /security/ldap # **create server** *server-name*

SSL을 활성화한 경우, 일반적으로 IP 주소 또는 FQDN인 *server-name*은 LDAP 서버의 보안 인증서에 있는 CN(Common Name)과 정확하게 일치해야 합니다. IP 주소가 지정되지 않았다면 DNS 서버를 구성해야 합니다.

단계 **4** (선택 사항) 사용자 역할 및 로케일에 대한 값을 저장하는 LDAP 속성을 설정합니다.
Firepower-chassis /security/ldap/server # **set attribute** *attr-name*

이 속성은 항상 이름값 쌍입니다. 시스템은 사용자 레코드를 쿼리하여 이 속성 이름과 일치하는 값을 찾습니다.

이 값은 기본 속성이 LDAP 제공자에 대해 설정되지 않은 경우 필요합니다.

단계 **5** (선택 사항) 원격 사용자가 로그인하고 시스템에서 사용자 이름에 기초한 사용자 DN을 얻으려고 시도할 때 서버에서 검색을 시도해야 하는 LDAP 계층 구조에서 특정한 고유 이름을 설정합니다.
Firepower-chassis /security/ldap/server # **set basedn** *basedn-name*

기본 DN 길이는 최대 255자에서 CN=username 길이를 뺀 값으로 설정할 수 있습니다. 여기서 username은 LDAP 인증을 사용하여 Firepower Chassis Manager 또는 FXOS CLI에 액세스하려는 원격 사용자를 나타냅니다.

이 값은 기본 DN의 기본값이 LDAP 제공자에 대해 설정되지 않은 경우 필요합니다.

단계 **6** (선택 사항) 기본 DN에 속하는 모든 객체에 대한 읽기 및 검색 권한이 있는 LDAP 데이터베이스 어카운트의 DN(고유 이름)을 설정합니다.
Firepower-chassis /security/ldap/server # **set binddn** *binddn-name*

지원되는 최대 문자열 길이는 ASCII 255자입니다.

단계 **7** (선택 사항) 정의된 필터와 일치하는 사용자 이름으로 LDAP 검색을 제한합니다.
Firepower-chassis /security/ldap/server # **set filter** *filter-value*

이 값은 기본 필터가 LDAP 제공자에 대해 설정되지 않은 경우 필요합니다.

단계 **8** Bind DN(바인드 DN) 필드에 지정된 LDAP 데이터베이스 어카운트의 비밀번호를 지정합니다.
Firepower-chassis /security/ldap/server # **set password**

공백, §(섹션 기호), ? (물음표) 또는 =(등호)를 제외한 모든 표준 ASCII 문자를 입력할 수 있습니다.

비밀번호를 설정하려면 **set password** 명령을 입력한 후 **Enter** 키를 누르고 프롬프트에 키 값을 입력합니다.

단계 **9** (선택 사항) Firepower eXtensible 운영 체제에서 사용자를 인증하기 위해 이 제공자를 사용하는 순서를 지정합니다.
Firepower-chassis /security/ldap/server # **set order** *order-num*

단계 **10** (선택 사항) LDAP 서버와의 통신에 사용되는 포트를 지정합니다. 표준 포트 번호는 389입니다.
Firepower-chassis /security/ldap/server # **set port** *port-num*

단계 **11** LDAP 서버와 통신할 때의 암호화 사용을 활성화 또는 비활성화합니다.
Firepower-chassis /security/ldap/server # **set ssl** {**yes**|**no**}

옵션은 다음과 같습니다.

- **yes**— 암호화가 필요합니다. 암호화를 협상할 수 없는 경우, 연결에 실패합니다.
- **no**— 암호화가 비활성화되어 있습니다. 인증 정보가 일반 텍스트로 전송됩니다.

LDAP은 STARTTLS를 사용합니다. 이는 포트 389를 사용하여 암호화된 통신을 허용합니다.

단계 **12** 시간이 초과되기 전에 시스템이 LDAP 데이터베이스에 연결을 시도하는 데 필요한 시간(초)을 지정합니다.
Firepower-chassis /security/ldap/server # **set timeout** *timeout-num*

1~60초의 정수를 입력하거나 0(숫자 0)을 입력하여 LDAP 제공자에 지정된 전역 시간 초과 값을 사용합니다. 기본값은 30초입니다.

단계 **13** LDAP 제공자 또는 서버 세부사항을 제공하는 벤더를 지정합니다.
Firepower-chassis /security/ldap/server # **set vendor** *{ms-ad | openldap}*

옵션은 다음과 같습니다.

- **ms-ad**— LDAP 제공자가 Microsoft Active Directory입니다.
- **openldap**— LDAP 제공자가 Microsoft Active Directory가 아닙니다.

단계 **14** (선택 사항) 인증 해지 목록 확인을 활성화합니다.
Firepower-chassis /security/ldap/server # **set revoke-policy** *{strict | relaxed}*

참고 이 컨피그레이션은 SSL 연결을 활성화한 경우에만 적용됩니다.

단계 **15** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/ldap/server # **commit-buffer**

다음의 예에서는 10.193.169.246이라는 이름의 LDAP 서버 인스턴스를 생성하고 binddn, 비밀번호, 순서, 포트, SSL 설정, 벤더 속성을 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap* # create server 10.193.169.246
Firepower-chassis /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap/server* # set password
Enter the password:
Confirm the password:
Firepower-chassis /security/ldap/server* # set order 2
Firepower-chassis /security/ldap/server* # set port 389
Firepower-chassis /security/ldap/server* # set ssl yes
Firepower-chassis /security/ldap/server* # set timeout 30
Firepower-chassis /security/ldap/server* # set vendor ms-ad
Firepower-chassis /security/ldap/server* # commit-buffer
Firepower-chassis /security/ldap/server #
```

다음의 예에서는 12:31:71:1231:45b1:0011:011:900이라는 이름의 LDAP 서버 인스턴스를 생성하고 binddn, 비밀번호, 순서, 포트, SSL 설정, 벤더 속성을 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap* # create server 12:31:71:1231:45b1:0011:011:900
Firepower-chassis /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap/server* # set password
Enter the password:
Confirm the password:
Firepower-chassis /security/ldap/server* # set order 1
Firepower-chassis /security/ldap/server* # set port 389
Firepower-chassis /security/ldap/server* # set ssl yes
Firepower-chassis /security/ldap/server* # set timeout 45
Firepower-chassis /security/ldap/server* # set vendor ms-ad
Firepower-chassis /security/ldap/server* # commit-buffer
Firepower-chassis /security/ldap/server #
```

LDAP 제공자 삭제

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 LDAP 모드를 시작합니다.
Firepower-chassis /security # **scope ldap**

단계 **3** 지정된 서버를 삭제합니다.
Firepower-chassis /security/ldap # **delete server** *serv-name*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/ldap # **commit-buffer**

다음 예에서는 ldap1이라는 LDAP 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap # delete server ldap1
Firepower-chassis /security/ldap* # commit-buffer
Firepower-chassis /security/ldap #
```

RADIUS 제공자 구성

RADIUS 제공자 속성 구성

이 작업에서 구성하는 속성은 이 유형의 모든 제공자 연결에 대한 기본 설정입니다. 개별 제공자에 이러한 속성의 설정이 포함되어 있는 경우에는 Firepower eXtensible 운영 체제에서 해당 설정을 사용하고 기본 설정을 무시합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 RADIUS 모드를 시작합니다.
Firepower-chassis /security # **scope radius**

단계 **3** (선택 사항) 서버가 다운되었다고 인지할 때까지 RADIUS 서버와의 통신을 재시도할 횟수를 지정합니다.
Firepower-chassis /security/radius # **set retries** *retry-num*

단계 **4** (선택 사항) 서버가 다운되었다고 인지할 때까지 시스템이 RADIUS 서버의 응답을 대기해야 하는 시간 간격을 설정합니다.
Firepower-chassis /security/radius # **set timeout** *seconds*

단계 **5** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/radius # **commit-buffer**

다음의 예에서는 RADIUS 재시도 횟수를 4로 설정하고 시간 초과 간격을 30초로 설정하며 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # set retries 4
Firepower-chassis /security/radius* # set timeout 30
Firepower-chassis /security/radius* # commit-buffer
Firepower-chassis /security/radius #
```

다음에 할 작업

RADIUS 제공자를 생성합니다.

RADIUS 제공자 생성

Firepower eXtensible 운영 체제는 최대 16개의 RADIUS 제공자를 지원합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 RADIUS 모드를 시작합니다.
Firepower-chassis /security # **scope radius**

단계 **3** RADIUS 서버 인스턴스를 생성하고 보안 RADIUS 서버 모드를 시작합니다.
Firepower-chassis /security/radius # **create server** *server-name*

단계 **4** (선택 사항) RADIUS 서버와의 통신에 사용되는 포트를 지정합니다.
Firepower-chassis /security/radius/server # **set authport** *authport-num*

단계 **5** RADIUS 서버 키를 설정합니다.
Firepower-chassis /security/radius/server # **set key**

키 값을 설정하려면 **set key** 명령을 입력한 후 **Enter** 키를 누르고 프롬프트에 키 값을 입력합니다.

단계 **6** (선택 사항) 이 서버에 시도할 순서를 지정합니다.
Firepower-chassis /security/radius/server # **set order** *order-num*

단계 **7** (선택 사항) 서버가 다운되었다고 인지할 때까지 RADIUS 서버와의 통신을 재시도할 횟수를 설정합니다.
Firepower-chassis /security/radius/server # **set retries** *retry-num*

단계 **8** 서버가 다운되었다고 인지할 때까지 시스템이 RADIUS 서버의 응답을 대기해야 하는 시간 간격을 지정합니다.
Firepower-chassis /security/radius/server # **set timeout** *seconds*

팁 RADIUS 제공자에 대해 2단계 인증을 선택하는 경우, 더 높은 **Timeout(**시간 초과**)** 값을 구성하는 것이 좋습니다.

단계 **9** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/radius/server # **commit-buffer**

다음 예에서는 radiusserv7이라는 이름의 서버 인스턴스를 생성하고 인증 포트를 5858로 설정하고 키를 radiuskey321로 설정하고 순서를 2로 설정하고 재시도 횟수를 4로 설정하며 시간 초과를 30으로 설정하고 2단계 인증을 활성화하며 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # create server radiusserv7
```

```
Firepower-chassis /security/radius/server* # set authport 5858
Firepower-chassis /security/radius/server* # set key
Enter the key: radiuskey321
Confirm the key: radiuskey321
Firepower-chassis /security/radius/server* # set order 2
Firepower-chassis /security/radius/server* # set retries 4
Firepower-chassis /security/radius/server* # set timeout 30
Firepower-chassis /security/radius/server* # commit-buffer
Firepower-chassis /security/radius/server #
```

RADIUS 제공자 삭제

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 RADIUS 모드를 시작합니다.
Firepower-chassis /security # **scope RADIUS**

단계 **3** 지정된 서버를 삭제합니다.
Firepower-chassis /security/radius # **delete server** *serv-name*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/radius # **commit-buffer**

다음 예에서는 radius1이라는 RADIUS 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # delete server radius1
Firepower-chassis /security/radius* # commit-buffer
Firepower-chassis /security/radius #
```

TACACS+ 제공자 구성

TACACS+ 제공자 속성 구성

이 작업에서 구성하는 속성은 이 유형의 모든 제공자 연결에 대한 기본 설정입니다. 개별 제공자에 이러한 속성의 설정이 포함되어 있는 경우에는 Firepower eXtensible 운영 체제에서 해당 설정을 사용하고 기본 설정을 무시합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 TACACS+ 모드를 시작합니다.

Firepower-chassis /security # **scope tacacs**

단계 **3** (선택 사항) 서버가 다운되었다고 인지할 때까지 시스템이 TACACS+ 서버의 응답을 대기해야 하는 시간 간격을 설정합니다.
Firepower-chassis /security/tacacs # **set timeout** *seconds*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/tacacs # **commit-buffer**

다음 예에서는 TACACS+ 시간 초과 간격을 45초로 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope tacacs
Firepower-chassis /security/tacacs # set timeout 45
Firepower-chassis /security/tacacs* # commit-buffer
Firepower-chassis /security/tacacs #
```

다음에 할 작업

TACACS+ 제공자를 생성합니다.

TACACS+ 제공자 생성

Firepower eXtensible 운영 체제는 최대 16개의 TACACS+ 제공자를 지원합니다.

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 TACACS+ 모드를 시작합니다.
Firepower-chassis /security # **scope tacacs**

단계 **3** TACACS+ 서버 인스턴스를 생성하고 보안 TACACS+ 서버 모드를 시작합니다.
Firepower-chassis /security/tacacs # **create server** *server-name*

단계 **4** TACACS+ 서버 키를 지정합니다.
Firepower-chassis /security/tacacs/server # **set key**

키 값을 설정하려면 **set key** 명령을 입력한 후 **Enter** 키를 누르고 프롬프트에 키 값을 입력합니다.

단계 **5** (선택 사항) 이 서버에 시도할 순서를 지정합니다.
Firepower-chassis /security/tacacs/server # **set order** *order-num*

단계 **6** 서버가 다운되었다고 인지할 때까지 시스템이 TACACS+ 서버의 응답을 대기해야 하는 시간 간격을 지정합니다.
Firepower-chassis /security/tacacs/server # **set timeout** *seconds*

팁 TACACS+ 제공자에 대한 2단계 인증을 선택한 경우, 더 높은 Timeout(시간 초과) 값을 구성하는 것이 좋습니다.

단계 **7** (선택 사항) TACACS+ 서버와의 통신에 사용되는 포트를 지정합니다.

Firepower-chassis /security/tacacs/server # **set port** *port-num*

단계 **8** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/tacacs/server # **commit-buffer**

다음 예에서는 tacacsserv680이라는 이름의 서버 인스턴스를 생성하고, 키는 tacacskey321로, 순서는 4로, 인증 포트는 5859로 설정한 다음, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope tacacs
Firepower-chassis /security/tacacs # create server tacacsserv680
Firepower-chassis /security/tacacs/server* # set key
Enter the key: tacacskey321
Confirm the key: tacacskey321
Firepower-chassis /security/tacacs/server* # set order 4
Firepower-chassis /security/tacacs/server* # set port 5859
Firepower-chassis /security/tacacs/server* # commit-buffer
Firepower-chassis /security/tacacs/server #
```

TACACS+ 제공자 삭제

절차

단계 **1** 보안 모드를 시작합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 TACACS+ 모드를 시작합니다.
Firepower-chassis /security # **scope tacacs**

단계 **3** 지정된 서버를 삭제합니다.
Firepower-chassis /security/tacacs # **delete server** *serv-name*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /security/tacacs # **commit-buffer**

다음 예에서는 tacacs1이라는 TACACS+ 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope tacacs
Firepower-chassis /security/tacacs # delete server tacacs1
Firepower-chassis /security/tacacs* # commit-buffer
Firepower-chassis /security/tacacs #
```

Syslog 구성

시스템 로깅은 디바이스의 메시지를 syslog 데몬을 실행 중인 서버로 수집하는 방식입니다. 중앙 syslog 서버에 로깅하면 로그와 알림을 집계하는 데 도움이 됩니다. syslog 서비스는 메시지를 수신하고 파일로 저장하거나 간단한 컨피그레이션 파일에 따라 인쇄합니다. 이 로깅 양식을 통해 로그를 안전하게 장기 보관할 수 있습니다. 로그는 일상적인 문제 해결과 사고 처리에 모두 유용합니다.

절차

단계 **1** 모니터링 모드를 시작합니다.
Firepower-chassis# **scope monitoring**

단계 **2** syslogs의 콘솔 전송을 활성화하거나 비활성화합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog console**

단계 **3** (선택 사항) 사용자가 표시하려는 가장 낮은 메시지 레벨을 선택합니다. syslog가 활성화된 경우 콘솔에 해당 레벨 이상의 메시지가 표시됩니다. 레벨 옵션은 내림차순으로 긴급도를 나열합니다. 기본 레벨은 Critical(위험)입니다.
Firepower-chassis /monitoring # **set syslog console level** {**emergencies** | **alerts** | **critical**}

단계 **4** 운영 체제별로 syslog 정보의 모니터링을 활성화하거나 비활성화합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog monitor**

단계 **5** (선택 사항) 사용자가 표시하려는 가장 낮은 메시지 레벨을 선택합니다. 모니터 상태가 활성화된 경우, 해당 레벨 이상의 메시지가 표시됩니다. 레벨 옵션은 내림차순으로 긴급도를 나열합니다. 기본 레벨은 Critical(위험)입니다.
Firepower-chassis /monitoring # **set syslog monitor level** {**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

참고 Critical(위험) 미만 레벨의 메시지는 **terminal monitor** 명령을 입력한 경우에만 터미널 모니터에 표시됩니다.

단계 **6** syslog 정보를 syslog 파일에 쓰는 기능을 활성화하거나 비활성화합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog file**

단계 **7** 메시지가 기록된 파일 이름을 지정합니다. 파일 이름에는 최대 16자를 사용할 수 있습니다.
Firepower-chassis /monitoring # **set syslog file name** *filename*

단계 **8** (선택 사항) 파일에 저장할 가장 낮은 메시지 레벨을 선택합니다. 파일 상태가 활성화된 경우, syslog 파일에 해당 레벨 이상의 메시지가 저장됩니다. 레벨 옵션은 내림차순으로 긴급도를 나열합니다. 기본 레벨은 Critical(위험)입니다.
Firepower-chassis /monitoring # **set syslog file level** {**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

단계 **9** (선택 사항) 시스템이 최신 메시지로 가장 오래된 메시지를 덮어쓰기 전에 최대 파일 크기(단위: 바이트)를 지정합니다. 범위는 4096~4194304바이트입니다.
Firepower-chassis /monitoring # **set syslog file size** *filesize*

단계 **10** 최대 3개의 외부 syslog 서버에 syslog 메시지를 전송하도록 구성합니다.

a) 최대 3개의 외부 syslog 서버에 syslog 메시지를 전송하는 기능을 활성화하거나 비활성화합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog remote-destination** {**server-1** | **server-2** | **server-3**}

b) (선택 사항) 외부 로그에 저장할 가장 낮은 메시지 레벨을 선택합니다. 원격 대상이 활성화된 경우, 외부 서버에 해당 레벨 이상의 메시지가 전송됩니다. 레벨 옵션은 내림차순으로 긴급도를 나열합니다. 기본 레벨은 Critical(위험)입니다.
Firepower-chassis /monitoring # **set syslog remote-destination** {**server-1** | **server-2** | **server-3**} **level**{**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

c) 지정된 원격 syslog 서버의 호스트 이름 또는 IP 주소를 지정합니다. 호스트 이름에는 최대 256자를 사용할 수 있습니다.
Firepower-chassis /monitoring # **set syslog remote-destination** {**server-1** | **server-2** | **server-3**} **hostname** *hostname*

d) (선택 사항) 지정된 원격 syslog 서버로 전송된 syslog 메시지에 포함된 기능 레벨을 지정합니다.
Firepower-chassis /monitoring # **set syslog remote-destination** {**server-1** | **server-2** | **server-3**} **facility** {**local0** | **local1** | **local2** | **local3** | **local4** | **local5** | **local6** | **local7**}

단계 **11** 로컬 소스를 구성합니다. 활성화하거나 비활성화하려는 각 로컬 소스에 대해 다음 명령을 입력합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog source** {**audits** | **events** | **faults**}

다음 중 하나일 수 있습니다.

- **audits(**감사**)** — 모든 감사 로그 이벤트 로깅을 활성화 또는 비활성화합니다.
- **events(**이벤트**)** — 모든 시스템 이벤트 기록을 활성화 또는 비활성화합니다.
- **faults(**결함**)** — 모든 시스템 결함 로깅을 활성화 또는 비활성화합니다.

단계 **12** 트랜잭션을 커밋합니다.
Firepower-chassis /monitoring # **commit-buffer**

이 예에서는 로컬 파일에서 syslog 메시지의 스토리지를 활성화하는 방법을 보여주며 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # disable syslog console
Firepower-chassis /monitoring* # disable syslog monitor
Firepower-chassis /monitoring* # enable syslog file
Firepower-chassis /monitoring* # set syslog file name SysMsgsFirepower
Firepower-chassis /monitoring* # set syslog file level notifications
Firepower-chassis /monitoring* # set syslog file size 4194304
Firepower-chassis /monitoring* # disable syslog remote-destination server-1
Firepower-chassis /monitoring* # disable syslog remote-destination server-2
Firepower-chassis /monitoring* # disable syslog remote-destination server-3
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

DNS 서버 구성

시스템에서 호스트 이름-IP 주소를 확인해야 하는 경우 DNS 서버를 지정해야 합니다. 예를 들어 DNS 서버를 구성하지 않으면 Firepower 섀시에서 설정을 구성할 때 www.cisco.com 등의 이름을 사용할 수 없습니다. 서버의 IP 주소(IPv4 또는 IPv6 주소 중 하나가 될 수 있음)를 사용해야 합니다. 최대 4개까지 DNS 서버를 구성할 수 있습니다.

참고 여러 DNS 서버를 구성할 때 시스템에서는 임의의 순서로 서버만 검색합니다. 로컬 관리 명령에 DNS 서버 조회가 필요한 경우, 임의의 순서로 3개의 DNS 서버만 검색할 수 있습니다.

절차

단계 **1** 시스템 모드를 시작합니다.
Firepower-chassis #**scope system**

단계 **2** 시스템 서비스 모드를 시작합니다.
Firepower-chassis /system #**scope services**

단계 **3** DNS 서버를 생성하거나 삭제하려면 다음과 같이 적절한 명령을 입력합니다.

- 지정된 IPv4 또는 IPv6 주소를 사용하는 DNS 서버를 사용하도록 시스템을 구성하려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **createdns** *{ip-addr | ip6-addr}*

- 지정된 IPv4 또는 IPv6 주소를 사용하는 DNS 서버를 삭제하려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **deletedns** *{ip-addr | ip6-addr}*

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower /system/services # **commit-buffer**

다음 예에서는 IPv4 주소 192.168.200.105를 사용하는 DNS 서버를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create dns 192.168.200.105
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

다음 예에서는 IPv6 주소 2001:db8::22:F376:FF3B:AB3F를 사용하는 DNS 서버를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create dns 2001:db8::22:F376:FF3B:AB3F
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

다음 예에서는 IP 주소 192.168.200.105를 사용하는 DNS 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete dns 192.168.200.105
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

10 장

인터페이스 관리

Firepower Security Appliance 인터페이스 정보

Firepower 4100/9300 섀시는 단일 인터페이스뿐만 아니라 EtherChannel(포트 채널) 인터페이스도 지원합니다. EtherChannel 인터페이스는 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

인터페이스 유형

각 인터페이스는 다음 유형 중 하나일 수 있습니다.

- Data(데이터)(기본값) -- 데이터 인터페이스는 논리적 디바이스 간에 공유할 수 없습니다.
- Management(관리) -- 관리 인터페이스는 논리적 디바이스 간에 공유할 수 있습니다. 논리적 디바이스당 관리 인터페이스 1개만 할당할 수 있습니다.

  Firepower Threat Defense 애플리케이션 내에서 물리적 관리 인터페이스는 논리적 진단 인터페이스와 논리적 관리 인터페이스 간에 공유됩니다. 논리적 관리 인터페이스는 디바이스에 있는 다른 인터페이스와 분리되어 있습니다. 이 인터페이스는 디바이스를 Firepower Management Center에 설치하고 등록하는 데 사용됩니다. 또한, 별도의 SSH 서버를 실행하며 자체 로컬 인증, IP 주소 및 정적 라우팅을 사용합니다. CLI에서 **configure network** 명령을 사용하여 설정을 구성하고 Management Center**Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Devices(**디바이스**)** > **Management(**관리**)** 영역에서 IP 주소를 변경할 수 있습니다.

논리적 진단 인터페이스는 Management Center **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Interfaces(**인터페이스**)** 화면에서 나머지 데이터 인터페이스와 함께 구성할 수 있습니다. 진단 인터페이스 사용은 선택 사항입니다. 진단 인터페이스 및 데이터 인터페이스는 LDAP 또는 RADIUS 외부 인증을 허용합니다. 예를 들어, 데이터 인터페이스에서 SSH 액세스를 허용하지 않으려면 SSH 액세스에 대해 진단 인터페이스를 구성할 수 있습니다. 진단 인터페이스는 관리 트래픽만 허용하며 통과 트래픽은 허용하지 않습니다.

- Firepower-eventing(Firepower 이벤트) -- 이 인터페이스는 Firepower Threat Defense 디바이스의 보조 관리 인터페이스입니다. 이 인터페이스를 사용하려면 Firepower Threat Defense CLI에서 해당 IP 주소 및 기타 파라미터를 구성해야 합니다. 예를 들면 관리 트래픽을 이벤트(예: 웹 이벤트)에서 분리할 수 있습니다. Firepower Management Center 명령 참조에서 **configure network** 명령을 참조하십시오.
- Cluster(클러스터) -- 클러스터링된 논리적 디바이스에 사용되는 특수 인터페이스 유형입니다. 이 유형은 유닛 클러스터 간 통신을 지원하는 클러스터 제어 링크에 자동으로 할당됩니다. 기본적으로 클러스터 제어 링크는 포트 채널 48에서 자동으로 생성됩니다.

참고 Firepower Management Center 또는 Firepower Threat Defense CLI를 사용하여 두 개의 업링크, 브레이크아웃 또는 데이터 포트 인터페이스를 인라인 쌍으로 구성할 수 있습니다. 두 포트가 인라인 쌍으로 구성되면 단일 인터페이스로 작동합니다. 그러면 이 컨피그레이션이 FXOS 섀시로 전파됩니다.

인라인 쌍에는 다음과 같은 제한 사항이 있습니다.

- 두 개의 포트 인터페이스는 고유해야 합니다. 포트는 하나의 인라인 쌍에 조인하면 다른 인라인 쌍에 조인할 수 없습니다.
- 업링크 포트, 데이터 포트 또는 브레이크아웃 포트만 인라인 쌍으로 구성할 수 있습니다.

자세한 내용은 Firepower Management Center 컨피그레이션 가이드의 IPS 전용 인터페이스의 인라인 설정 구성 항목을 참조하십시오.

하드웨어 바이패스 쌍

Firepower Threat Defense의 경우, Firepower 9300 및 4100 Series에서 특정 인터페이스 모듈을 사용하면 하드웨어 바이패스 기능을 활성화할 수 있습니다. 하드웨어 바이패스는 정전 중에도 인라인 인터페이스 쌍 사이에서 트래픽이 계속 통과하게 해줍니다. 이 기능은 소프트웨어 또는 하드웨어 장애가 발생할 경우 네트워크 연결을 유지하는 데 사용될 수 있습니다.

하드웨어 바이패스 기능은 하드웨어 바이패스 애플리케이션 내에 구성됩니다. 이러한 인터페이스는 하드웨어 바이패스 쌍으로 사용할 필요가 없으며, ASA와 Firepower Threat Defense 애플리케이션 둘 다에 대해 일반 인터페이스로 사용할 수 있습니다. 하드웨어 바이패스 지원 인터페이스는 브레이크아웃 포트용으로 구성할 수 없습니다. 하드웨어 바이패스 기능을 사용하려는 경우, 포트를 EtherChannel로 구성하지 마십시오. 그렇지 않으면 이러한 인터페이스를 일반 인터페이스 모드에서 EtherChannel 멤버로 포함할 수 있습니다.

Firepower Threat Defense는 다음 모델에서 특정 네트워크 모듈의 인터페이스 쌍에 대해 하드웨어 바이패스를 지원합니다.

- Firepower 9300
- Firepower 4100 Series

이러한 모델에 대해 지원되는 하드웨어 바이패스 네트워크 모듈은 다음과 같습니다.

- Firepower 6 포트 1G SX FTW Network Module single-wide(FPR-NM-6X1SX-F)
- Firepower 6 포트 10G SR FTW Network Module single-wide(FPR-NM-6X10SR-F)
- Firepower 6 포트 10G LR FTW Network Module single-wide(FPR-NM-6X10LR-F)
- Firepower 2 포트 40G SR FTW Network Module single-wide(FPR-NM-2X40G-F)
- Firepower 8포트 1G Copper FTW Network Module single-wide(FPR-NM-8X1G-F)

하드웨어 바이패스에서는 다음 포트 쌍만 사용할 수 있습니다.

- 1 및 2
- 3 및 4
- 5 및 6
- 7 및 8

점보 프레임 지원

Firepower 4100/9300 섀시는 기본적으로 활성화되어 있는 점보 프레임을 지원합니다. Firepower 4100/9300 섀시에 설치되어 있는 특정한 논리적 디바이스에서 점보 프레임 지원을 활성화하려면 논리적 디바이스에서 인터페이스에 적절한 MTU 설정을 구성해야 합니다.

Firepower 4100/9300 섀시에 있는 애플리케이션에 대해 지원되는 최대 MTU는 9184입니다.

인터페이스 속성 편집

절차

단계 **1** 인터페이스 모드를 시작합니다.
scopeeth-uplink
scope fabric a

단계 **2** 인터페이스를 활성화합니다.
enterinterface *interface_id*
enable

예제:

```
Firepower /eth-uplink/fabric # enter interface Ethernet1/8
Firepower /eth-uplink/fabric/interface # enable
```

참고 이미 포트 채널의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. 포트 채널의 멤버인 인터페이스에서 **enter interface** 또는 **scope interface** 명령을 사용하는 경우 객체가 존재하지 않음을 알리는 오류가 표시됩니다. 포트 채널에 인터페이스를 추가하기 전에 **enter interface** 명령을 사용하여 인터페이스를 편집해야 합니다.

단계 **3** (선택 사항) 인터페이스 유형을 설정합니다.
setport-type {**data** | **firepower-eventing** | **mgmt** | **cluster**}

예제:

```
Firepower /eth-uplink/fabric/interface # set port-type mgmt
```

data 키워드는 기본 유형입니다. **cluster** 키워드는 선택하지 마십시오.

단계 **4** (선택 사항) 인터페이스 속도를 설정합니다.
setadmin-speed {**10gbps** | **1gbps**}

예제:

```
Firepower /eth-uplink/fabric/interface* # set admin-speed 1gbps
```

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

포트 채널 생성

EtherChannel(port-channel)은 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

Firepower 4100/9300 섀시에서 EtherChannel을 만들면, 물리적 링크가 작동 중이더라도 EtherChannel은 물리적 디바이스에 할당될 때까지 **Suspended(**일시 중단**)** 상태로 유지됩니다. 다음의 상황에서는 EtherChannel의 **Suspended(**일시 중단**)** 상태가 해제됩니다.

- EtherChannel은 독립형 논리적 디바이스에 대한 데이터 또는 관리 포트로 추가됩니다.
- EtherChannel은 클러스터의 일부인 논리적 디바이스에 대한 관리 또는 CCL 포트로 추가됩니다.
- EtherChannel은 클러스터의 일부인 논리적 디바이스에 대한 데이터 포트로 추가되며, 하나 이상의 보안 모듈이 클러스터에 조인됩니다.

EtherChannel은 논리적 디바이스에 할당될 때까지 나타나지 않습니다. 논리적 디바이스에서 EtherChannel이 제거되거나 논리적 디바이스가 삭제된 경우, EtherChannel은 **Suspended(**일시 중단**)** 상태로 되돌아갑니다.

시작하기 전에

Firepower 4100/9300 섀시에서는 활성 LACP(Link Aggregation Control Protocol) 모드에서 EtherChannel만 지원합니다. 최고의 호환성을 위해 연결 스위치 포트를 Active(활성) 모드로 설정하는 것이 좋습니다.

절차

단계 **1** 인터페이스 모드를 시작합니다.
scopeeth-uplink
scope fabric a

단계 **2** 포트 채널을 생성합니다.
createport-channel *id*
enable

단계 **3** 멤버 인터페이스를 할당합니다.
createmember-port *interface_id*

예제:

```
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
Firepower /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
Firepower /eth-uplink/fabric/port-channel/member-port* # exit
```

단계 **4** (선택 사항) 인터페이스 유형을 설정합니다.
setport-type {**data** | **mgmt** | **cluster**}

예제:

```
Firepower /eth-uplink/fabric/port-channel # set port-type mgmt
```

data 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

단계 **5** (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.
setspeed {**10gbps** | **1gbps**}

예제:

```
Firepower /eth-uplink/fabric/port-channel* # set speed 1gbps
```

단계 **6** 컨피그레이션을 커밋합니다.
commit-buffer

브레이크아웃 케이블 구성

다음 절차에서는 Firepower 4100/9300 섀시와 함께 사용할 브레이크아웃 케이블을 구성하는 방법을 보여줍니다. 브레이크아웃 케이블을 사용하여 1개의 40Gbps 포트 대신 4개의 10Gbps 포트를 제공할 수 있습니다.

시작하기 전에

하드웨어 바이패스 지원 인터페이스는 브레이크아웃 포트용으로 구성할 수 없습니다.

절차

단계 **1** 다음 명령을 사용하여 새 브레이크아웃 케이블을 생성합니다.

a) 케이블링 모드를 시작합니다.
scopecabling
scope fabric a

b) 브레이크아웃을 생성합니다.
createbreakout *network_module_slot port*

예제:

```
Firepower /cabling/fabric/ # create breakout 2 1
```

c) 컨피그레이션을 커밋합니다.
commit-buffer

자동 재부팅이 수행됩니다. 브레이크아웃을 둘 이상 구성하는 경우 commit-buffer 명령을 실행하기 전에 모두 생성해야 합니다.

단계 **2** 다음 명령을 사용하여 브레이크아웃 포트를 활성화하고 구성합니다.

a) 인터페이스 모드를 시작합니다.
scopeeth-uplink

scopefabrica

scopeaggr-interface *network_module_slot port*

참고 이미 포트 채널의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. 포트 채널의 멤버인 인터페이스에서 **enter interface** 또는 **scope interface** 명령을 사용하는 경우 객체가 존재하지 않음을 알리는 오류가 표시됩니다. 포트 채널에 인터페이스를 추가하기 전에 **enter interface** 명령을 사용하여 인터페이스를 편집해야 합니다.

b) **set** 명령을 사용하여 인터페이스 속도 및 포트 유형을 구성합니다.
enable 또는 **disable** 명령을 사용하여 인터페이스의 관리 상태를 설정합니다.

c) 컨피그레이션을 커밋합니다.
commit-buffer

설치된 인터페이스 보기

섀시에 설치된 인터페이스의 상태를 보려면 다음 절차를 사용합니다.

절차

단계 **1** 인터페이스 모드를 시작합니다.
scopeeth-uplink
scope fabric a

단계 **2** 섀시에 설치된 인터페이스를 표시합니다.
showinterface

참고 포트 채널에서 포트 역할을 하는 인터페이스는 이 목록에 나타나지 않습니다.

```
Firepower /eth-uplink/fabric # show interface

Interface:
    Port Name       Port Type          Admin State Oper State       State Reason
    --------------- ------------------ ----------- ---------------- ------------
    Ethernet1/1     Mgmt               Enabled     Up
    Ethernet1/2     Data               Enabled     Link Down        Link failure
 or not-connected
    Ethernet1/3     Data               Enabled     Up
    Ethernet1/4     Data               Enabled     Sfp Not Present  Unknown
    Ethernet1/6     Data               Enabled     Sfp Not Present  Unknown
    Ethernet1/7     Data               Enabled     Sfp Not Present  Unknown
    Ethernet1/8     Data               Disabled    Sfp Not Present  Unknown
    Ethernet2/1     Data               Enabled     Up
    Ethernet2/2     Data               Enabled     Up
    Ethernet2/4     Data               Enabled     Up
    Ethernet2/5     Data               Enabled     Up
    Ethernet2/6     Data               Enabled     Up
    Ethernet3/2     Data               Enabled     Up
    Ethernet3/4     Data               Enabled     Up
```

11 장

논리적 디바이스

논리적 디바이스 정보

논리적 디바이스를 생성할 때 Firepower 4100/9300 섀시 수퍼바이저는 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 지정된 보안 모듈/엔진에 푸시하여 논리적 디바이스를 구축합니다. 섀시 내 클러스터의 경우에는 Firepower 섀시에 설치된 모든 보안 모듈에 푸시하여 구축합니다.

다음 2가지 유형의 논리적 디바이스 중 하나를 생성할 수 있습니다.

- 독립형 — Firepower 섀시에 설치된 각 보안 모듈/엔진에 독립형 논리적 디바이스를 생성할 수 있습니다.
- 클러스터 — 클러스터링으로 여러 개의 보안 모듈을 하나의 논리적 디바이스로 그룹화할 수 있습니다. 클러스터는 처리량 증대 및 여러 디바이스의 이중화라는 목표를 달성하는 동시에 단일 디바이스(관리, 네트워크에 통합)의 모든 편의성을 제공합니다. Firepower 9300과 같은 다중 모듈 디바이스는 섀시 내 클러스터링을 지원합니다.

참고 여러 보안 모듈을 지원하는 Firepower 4100/9300 섀시에는 독립형 또는 클러스터 중 한 가지 유형의 논리적 디바이스만 생성할 수 있습니다. 즉, 보안 모듈 3개가 설치된 경우, 보안 모듈 하나에 독립형 논리적 디바이스를 생성한 다음 나머지 논리적 디바이스 2개를 사용하여 클러스터를 생성할 수 없습니다.

참고 독립형 논리적 디바이스를 구성 중인 경우, 섀시에 있는 모든 모듈에 동일한 소프트웨어 유형을 설치해야 하며 다른 소프트웨어 유형은 현재 지원되지 않습니다. 모듈은 특정 디바이스 유형의 다른 버전을 실행할 수 있지만 모든 모듈은 동일한 유형의 논리적 디바이스로 구성되어야 합니다.

독립형 논리적 디바이스 생성

Firepower 섀시에 설치된 각 보안 모듈/엔진에 대해 독립형 논리적 디바이스를 생성할 수 있습니다.

독립형 **ASA** 논리적 디바이스 생성

Firepower 4100/9300 섀시에 설치된 각각의 보안 모듈/엔진에 독립형 논리적 디바이스를 생성할 수 있습니다. Firepower 9300과 같은 다중 모듈 디바이스에서는 클러스터가 구성되어 있는 경우 독립형 논리적 디바이스를 생성할 수 없습니다. 독립형 디바이스를 구성하려면 먼저 클러스터를 삭제해야 합니다.

참고 또는 서드파티 Radware DefensePro 가상 플랫폼을 보안 모듈의 ASA 방화벽보다 먼저 실행되는 DDoS 탐지 및 완화 서비스로 설치할 수 있습니다(서비스 체이닝 정보, 181 페이지 참조).

참고 하나의 섀시 내의 모든 모듈에 동일한 소프트웨어 유형을 설치해야 합니다. 다른 소프트웨어 유형은 지원되지 않습니다. 모듈은 특정 디바이스 유형의 다른 버전을 실행할 수 있지만 모든 모듈은 동일한 유형의 논리적 디바이스로 구성되어야 합니다.

시작하기 전에

- 논리적 디바이스에 사용할 보안 모듈/엔진에 이미 논리적 디바이스가 구성되어 있는 경우, 먼저 기존의 논리적 디바이스를 삭제해야 합니다(논리적 디바이스 삭제, 189 페이지 참조).
- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 54 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 다운로드합니다(Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 57 페이지 참조).
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다.
- Firepower 4100/9300 섀시에서 라우팅된 방화벽 모드 ASA만 구축할 수 있습니다.

절차

단계 **1** 보안 서비스 모드를 시작합니다.

Firepower# **scopessa**

단계 **2** 논리적 디바이스를 생성합니다.
Firepower /ssa # **createlogical-device** *device_name***asa** *slot_id***standalone**

단계 **3** 논리적 디바이스에 대한 설명을 입력합니다.
Firepower /ssa/logical-device* # **setdescription** *"logical device description"*

단계 **4** 논리적 디바이스에 관리 및 데이터 인터페이스를 할당합니다.
Firepower /ssa/logical-device* # **createexternal-port-link** *name interface_id***asa**
Firepower-chassis /ssa/logical-device/external-port-link* # **exit**

*name*은 Firepower 4100/9300 섀시 수퍼바이저에서 사용되며 보안 모듈 컨피그레이션에 사용되는 인터페이스 이름이 아닙니다. 인터페이스마다 단계를 반복합니다.

단계 **5** 관리 부트스트랩 정보를 구성합니다.

a) 부트스트랩 객체를 생성합니다.
Firepower /ssa/logical-device* # **createmgmt-bootstrapasa**

b) 비밀번호를 활성화합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-key-secretPASSWORD**

c) 비밀번호 값을 설정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # **setvalue**
값: *password*

d) 비밀번호 컨피그레이션 범위를 종료합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # **exit**

e) 관리 IP 주소를 구성합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createipv4** *slot_id***default**

f) 게이트웨이 주소를 설정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # **setgateway** *gateway_address*

g) IP 주소 및 마스크를 설정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # **setip** *ip_address***mask** *network_mask*

h) 관리 IP 컨피그레이션 범위를 종료합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # **exit**

i) 관리 부트스트랩 컨피그레이션 범위를 종료합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **exit**

단계 **6** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa # create logical-device MyDevice1 asa 1 standalone
```

```
Firepower /ssa/logical-device* # set description "logical device description"
Firepower /ssa/logical-device* # create external-port-link inside Ethernet1/1 asa
Firepower /ssa/logical-device/external-port-link* # set description "inside link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link management Ethernet1/7 asa
Firepower /ssa/logical-device/external-port-link* # set description "management link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link outside Ethernet1/2 asa
Firepower /ssa/logical-device/external-port-link* # set description "external link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value: <password>
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 default
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 1.1.1.254
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 1.1.1.1 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* # show configuration pending
+enter logical-device MyDevice1 asa 1 standalone
+    enter external-port-link inside Ethernet1/1 asa
+        set decorator ""
+        set description "inside link"
+    exit
+    enter external-port-link management Ethernet1/7 asa
+        set decorator ""
+        set description "management link"
+    exit
+    enter external-port-link outside Ethernet1/2 asa
+        set decorator ""
+        set description "external link"
+    exit
+    enter mgmt-bootstrap asa
+        enter bootstrap-key-secret PASSWORD
+            set value
+        exit
+        enter ipv4 1 default
+            set gateway 1.1.1.254
+            set ip 1.1.1.1 mask 255.255.255.0
+        exit
+    exit
+    set description "logical device description"
+exit
Firepower /ssa/logical-device* # commit-buffer
```

독립형 **Threat Defense** 논리적 디바이스 생성

Firepower 4100/9300 섀시에 설치된 각각의 보안 모듈/엔진에 독립형 논리적 디바이스를 생성할 수 있습니다. Firepower 9300과 같은 다중 모듈 디바이스에서는 클러스터가 구성되어 있는 경우 독립형 논리적 디바이스를 생성할 수 없습니다. 독립형 디바이스를 구성하려면 먼저 클러스터를 삭제해야 합니다.

참고 또는 서드파티 Radware DefensePro 가상 플랫폼을 보안 모듈의 Firepower Threat Defense 논리적 디바이스보다 먼저 실행되는 DDoS 탐지 및 완화 서비스로 설치할 수 있습니다(서비스 체이닝 정보, 181 페이지 참조).

참고 하나의 섀시 내의 모든 모듈에 동일한 소프트웨어 유형을 설치해야 합니다. 다른 소프트웨어 유형은 지원되지 않습니다. 모듈은 특정 디바이스 유형의 다른 버전을 실행할 수 있지만 모든 모듈은 동일한 유형의 논리적 디바이스로 구성되어야 합니다.

시작하기 전에

- 논리적 디바이스에 사용할 보안 모듈/엔진에 이미 논리적 디바이스가 구성되어 있는 경우, 먼저 기존의 논리적 디바이스를 삭제해야 합니다(논리적 디바이스 삭제, 189 페이지 참조).
- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 54 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 다운로드합니다(Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 57 페이지 참조).
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 최소 하나 이상의 데이터 유형 인터페이스도 구성해야 합니다. 또는 Firepower 이벤트 처리 인터페이스를 생성하여 모든 이벤트 트래픽(예: 웹 이벤트)을 전달할 수 있습니다.

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower# **scopessa**

단계 **2** 논리적 디바이스를 생성합니다.
Firepower /ssa # **createlogical-device** *device_name***ftd** *slot_id***standalone**

*device_name*은 Firepower 4100/9300 섀시 수퍼바이저가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 디바이스 이름이 아닙니다.

단계 **3** 논리적 디바이스에 관리 및 데이터 인터페이스를 할당합니다.
Firepower /ssa/logical-device* # **createexternal-port-link** *name interface_id***ftd**
Firepower-chassis /ssa/logical-device/external-port-link* # **exit**

*name*은 Firepower 4100/9300 섀시 수퍼바이저에서 사용되며 보안 모듈 컨피그레이션에 사용되는 인터페이스 이름이 아닙니다. 인터페이스마다 단계를 반복합니다.

단계 **4** 관리 부트스트랩 파라미터를 구성합니다.

a) 부트스트랩 객체를 생성합니다.
Firepower /ssa/logical-device* # **createmgmt-bootstrapftd**

b) 관리하는 Firepower Management Center의 IP 주소를 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-keyFIREPOWER_MANAGER_IP**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **set value** *IP_address*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **exit**

c) 논리적 디바이스가 작동할 모드(Routed(라우팅됨) 또는 Transparent(투명))를 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-keyFIREWALL_MODE**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **set value** *firewall_mode*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **exit**

d) 디바이스와 Firepower Management Center 간에 공유할 키를 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-key-secretREGISTRATION_KEY**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # **set value**

값: *registration_key*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # **exit**

e) 논리적 디바이스에 사용할 비밀번호를 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-key-secretPASSWORD**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # **set value**

값: *password*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # **exit**

f) 논리적 디바이스의 정규화된 호스트 이름을 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-keyFQDN**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **set value** *fqdn*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **exit**

g) 논리적 디바이스에서 사용할 쉼표로 구분된 DNS 서버 목록을 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-keyDNS_SERVERS**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **set value** *dns_servers*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **exit**

h) 논리적 디바이스를 위한 검색 도메인의 쉼표로 구분된 목록을 지정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* # **createbootstrap-keySEARCH_DOMAINS**
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **set value** *search_domains*

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # **exit**

i) 관리 인터페이스 설정을 구성합니다.
다음과 같이 IPv4 관리 인터페이스 객체를 생성합니다.

1 관리 인터페이스 객체를 생성합니다.

Firepower /ssa/logical-device/mgmt-bootstrap* # **createipv4** *slot_id***firepower**

2 게이트웨이 주소를 설정합니다.

Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # **setgateway** *gateway_address*

3 IP 주소 및 마스크를 설정합니다.

Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # **setip** *ip_address***mask** *network_mask*

4 관리 IP 컨피그레이션 범위를 종료합니다.

Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # **exit**

다음과 같이 IPv6 관리 인터페이스 객체를 생성합니다.

1 관리 인터페이스 객체를 생성합니다.

Firepower /ssa/logical-device/mgmt-bootstrap* # **createipv6** *slot_id***firepower**

2 게이트웨이 주소를 설정합니다.

Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # **setgateway** *gateway_address*

3 IP 주소 및 접두사를 설정합니다.

Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # **set ip** *ip_address***prefix-length** *prefix*

4 관리 IP 컨피그레이션 범위를 종료합니다.

Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # **exit**

j) 관리 부트스트랩 모드를 종료합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* #**exit**

단계 **5** 엔드 유저 라이선스 계약에 동의합니다.

a) Firepower /ssa/logical-device* # **exit**
b) Firepower /ssa* #**show app-instance**
Firepower Threat Defense 애플리케이션의 버전을 표시합니다.
c) Firepower /ssa* #**scope app ftd** *application_version*
d) Firepower /ssa/app* #**show license-agreement**
e) Firepower /ssa/app* #**accept-license-agreement**
f) Firepower /ssa/app* #**exit**

단계 **6** (선택 사항) Radware DefensePro 인스턴스를 설치합니다.
Firepower /ssa* # **scope slot** *slot_id*

Firepower/ssa/slot* # **createapp-instance vdp**

이 절차의 마지막 단계를 수행하여 논리적 디바이스 컨피그레이션을 커밋한 후, 계속해서 Radware DefensePro 데코레이터를 Firepower Threat Defense 논리적 디바이스로 서비스 체인에 구성해야 합니다. 독립형 논리적 디바이스에 Radware DefensePro 서비스 체인 구성, 182 페이지 절차를 4단계부터 참조하십시오. Firepower 4110 및 4120의 경우, 논리적 디바이스 컨피그레이션을 커밋하기 전에 앱 인스턴스를 생성해야 합니다.

단계 **7** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa #create logical-device MyDevice1 ftd 1 standalone
Firepower /ssa/logical-device* # create external-port-link inside Ethernet1/1 ftd
Firepower /ssa/logical-device/external-port-link* # set description "inside link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link management Ethernet1/7 ftd
Firepower /ssa/logical-device/external-port-link* # set description "management link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link outside Ethernet1/2 ftd
Firepower /ssa/logical-device/external-port-link* # set description "external link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create mgmt-bootstrap ftd
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key FIREPOWER_MANAGER_IP
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 10.0.0.100
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key FIREWALL_MODE
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value routed
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret REGISTRATION_KEY
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value:
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value:
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.31 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key FQDN
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value ftd.cisco.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key DNS_SERVERS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 192.168.1.1
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key SEARCH_DOMAINS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value search.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* # exit
Firepower /ssa* # scope app ftd 6.0.0.837
Firepower /ssa/app* # accept-license-agreement
Firepower /ssa/app* # commit-buffer
```

클러스터 구축

클러스터링을 사용하면 여러 개의 디바이스를 하나의 논리적 디바이스로 그룹화할 수 있습니다. 클러스터는 처리량 증대 및 여러 디바이스의 이중화라는 목표를 달성하는 동시에 단일 디바이스(관리, 네트워크에 통합)의 모든 편의성을 제공합니다. 여러 개의 모듈을 포함하는 Firepower 9300은 단일 섀시 내의 모든 모듈을 클러스터로 그룹화하는 섀시 내 클러스터링을 지원합니다. 또한, 여러 섀시를 함께 그룹화하는 섀시 간 클러스터링을 사용할 수 있습니다(Firepower 4100 Series 같은 단일 모듈 디바이스의 경우 섀시 간 클러스터링은 유일한 옵션임).

클러스터링 정보 - Firepower 4100/9300 섀시

클러스터는 하나의 논리적 유닛으로 작동하는 여러 개의 디바이스로 구성됩니다. Firepower 4100/9300 섀시에서 클러스터를 구축하려면 다음 작업을 수행합니다.

- 유닛 간 통신에 사용되는 클러스터 제어 링크(기본적으로, 포트 채널 48)를 생성합니다. 섀시 내 클러스터링(Firepower 9300 전용)의 경우, 이 링크는 클러스터 통신에 Firepower 9300 백플레인을 활용합니다. 섀시 간 클러스터링의 경우, 섀시 간의 통신을 위해 물리적 인터페이스를 이 EtherChannel에 수동으로 할당해야 합니다.
- 애플리케이션 내부에 클러스터 부트스트랩 컨피그레이션을 생성합니다.

  클러스터를 구축할 때, Firepower 4100/9300 섀시 수퍼바이저는 클러스터 이름, 클러스터 제어 링크 인터페이스 및 기타 클러스터 설정을 포함하는 각 유닛에 최소한의 부트스트랩 컨피그레이션을 푸시합니다. 클러스터링 환경을 맞춤 설정하려는 경우, 사용자가 일부 부트스트랩 컨피그레이션을 애플리케이션 내부에 구성할 수 있습니다.

- 데이터 인터페이스를 *Spanned* 인터페이스로 클러스터에 할당합니다.

  섀시 내 클러스터링의 경우, Spanned 인터페이스는 섀시 간 클러스터링과 마찬가지로 EtherChannel에 국한되지 않습니다. Firepower 9300 수퍼바이저는 EtherChannel 기술을 내부에 사용하여 트래픽을 공유 인터페이스의 다중 모듈에 로드 밸런싱하므로 모든 데이터 인터페이스 유형이 Spanned 모드에서 작동합니다. 섀시 간 클러스터링의 경우, 모든 데이터 인터페이스에 Spanned EtherChannel을 사용해야 합니다.

  참고 개별 인터페이스는 지원되지 않습니다(관리 인터페이스는 제외).

- 관리 인터페이스를 클러스터의 모든 유닛에 할당합니다.

다음 섹션에서는 클러스터링 개념 및 구현에 대한 자세한 정보를 제공합니다.

기본 유닛 및 보조 유닛 역할

클러스터의 멤버 중 하나는 기본 유닛입니다. 기본 유닛은 자동으로 결정됩니다. 그 외 멤버는 모두 보조 유닛입니다.

모든 컨피그레이션은 기본 유닛에서만 수행해야 하며, 컨피그레이션은 이후에 보조 유닛에 복제됩니다.

클러스터 제어 링크

클러스터 제어 링크는 포트 채널 48 인터페이스를 사용하여 자동으로 생성됩니다. 섀시 내 클러스터링의 경우, 이 인터페이스에는 멤버 인터페이스가 없습니다. 섀시 간 클러스터링의 경우, EtherChannel에 인터페이스를 하나 이상 추가해야 합니다. 이 클러스터 유형 EtherChannel은 섀시 내 클러스터링을 위한 클러스터 통신에 Firepower 9300 백플레인을 활용합니다.

2-멤버 섀시 간 클러스터의 경우 클러스터 제어 링크를 하나의 섀시에서 다른 섀시로 직접 연결하지 마십시오. 인터페이스에 직접 연결할 경우, 유닛 하나에 오류가 발생하면 클러스터 제어 링크에도 오류가 발생하므로 나머지 정상 유닛에도 오류가 발생합니다. 스위치를 통해 클러스터 제어 링크를 연결할 경우 클러스터 제어 링크는 가동 상태를 유지하여 정상 유닛을 지원합니다.

클러스터 제어 링크 트래픽에는 제어 및 데이터 트래픽이 모두 포함됩니다.

섀시 간 클러스터링을 위한 클러스터 제어 링크 크기 조정

가능한 경우, 각 섀시의 예상 처리량에 맞게 클러스터 제어 링크의 크기를 조정하여 클러스터 제어 링크가 최악의 시나리오를 처리할 수 있게 해야 합니다.

클러스터 제어 링크 트래픽은 주로 상태 업데이트 및 전달된 패킷으로 구성되어 있습니다. 클러스터 제어 링크의 트래픽 양은 언제든지 달라질 수 있습니다. 전달된 트래픽의 양은 로드 밸런싱 효율성 또는 중앙 집중식 기능에 많은 트래픽이 있는지에 따라 좌우됩니다. 예를 들면 다음과 같습니다.

- NAT의 경우 연결의 로드 밸런싱이 저하되며, 모든 반환 트래픽을 올바른 유닛으로 다시 밸런싱해야 합니다.

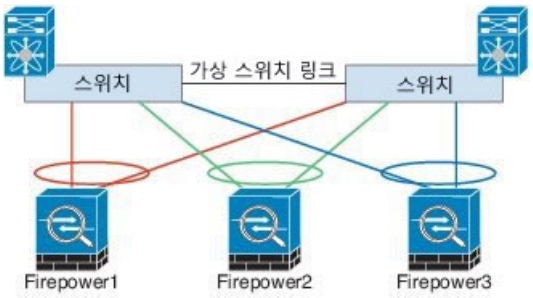
- 멤버가 변경된 경우, 클러스터에서는 다량의 연결을 다시 밸런싱해야 하므로 일시적으로 많은 양의 클러스터 제어 링크 대역폭을 사용합니다.

대역폭이 높은 클러스터 제어 링크를 사용하면 멤버가 변경될 경우 클러스터를 더 빠르게 통합할 수 있고 처리량 병목 현상을 방지할 수 있습니다.

참고 클러스터에 비대칭(다시 밸런싱된) 트래픽이 많은 경우 클러스터 제어 링크 크기를 늘려야 합니다.

섀시 간 클러스터링을 위한 클러스터 제어 링크 이중화

다음 다이어그램에는 VSS(Virtual Switching System) 또는 vPC(Virtual Port Channel) 환경에서 EtherChannel을 클러스터 제어 링크로 사용하는 방법이 나와 있습니다. EtherChannel의 모든 링크가 활성화되어 있습니다. 스위치가 VSS 또는 vPC의 일부일 경우 동일한 EtherChannel 내에서 Firepower 4100/9300 섀시 인터페이스를 연결하여 VSS 또는 vPC의 스위치와 별도로 분리할 수 있습니다. 이러한 별도의 스위치는 단일 스위치 역할을 수행하므로, 스위치 인터페이스는 동일한 EtherChannel 포트 채널 인터페이스의 멤버입니다. 이러한 EtherChannel은 디바이스 로컬이 아닌 Spanned EtherChannel입니다.



섀시 간 클러스터링을 위한 클러스터 제어 링크 안정성

클러스터 제어 링크 기능을 보장하려면 유닛 간의 RTT(round-trip time)가 20ms 미만이어야 합니다. 이러한 최대 레이턴시는 서로 다른 지리적 사이트에 설치된 클러스터 멤버와의 호환성을 개선합니다. 레이턴시를 확인하려면 유닛 간의 클러스터 제어 링크에서 ping을 수행합니다.

클러스터 제어 링크는 오류가 나거나 삭제된 패킷이 없는 안정적인 상태여야 합니다. 예를 들어, 사이트 간 구축 시에는 전용 링크를 사용해야 합니다.

클러스터 제어 링크 네트워크

Firepower 4100/9300 섀시는 섀시 ID와 슬롯 ID(127.2.*chassis_id*.*slot_id*)를 기반으로 하는 각 유닛에 대해 클러스터 제어 링크 인터페이스 IP 주소를 자동으로 생성합니다. 이 IP 주소는 FXOS에서 또는 애플리케이션 내에서 수동으로 설정할 수 없습니다. 클러스터 제어 링크 네트워크에서는 유닛 간에 라

우터가 포함될 수 없으며, 레이어 2 스위칭만 허용됩니다. 사이트 간 트래픽의 경우, OTV(Overlay Transport Virtualization)를 사용하는 것이 좋습니다.

관리 네트워크

모든 유닛을 단일 관리 네트워크에 연결하는 것이 좋습니다. 이러한 네트워크는 클러스터 제어 링크와 분리되어 있습니다.

관리 인터페이스

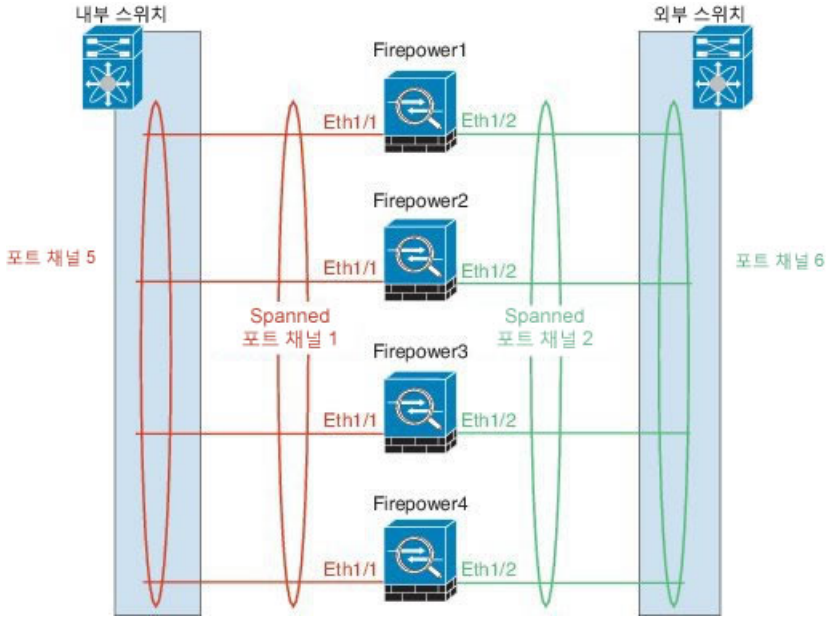
클러스터에 관리 유형 인터페이스를 할당할 수 있습니다. 이 인터페이스는 Spanned 인터페이스와는 다른 특수 개별 인터페이스입니다. 관리 인터페이스를 사용하면 각 유닛에 직접 연결할 수 있습니다.

ASA의 경우, 기본 클러스터 IP 주소는 현재 기본 유닛에 항상 속해 있는 클러스터를 위한 고정 주소입니다. 주소의 범위를 구성하여 현재 기본 유닛을 비롯한 각 유닛에서 해당 범위의 로컬 주소를 사용할 수 있도록 합니다. 기본 클러스터 IP 주소에서는 주소에 대한 일관된 관리 액세스를 제공합니다. 기본 유닛이 변경될 경우 기본 클러스터 IP 주소는 새 기본 유닛으로 이동되므로 클러스터는 지속적으로 원활하게 관리됩니다. 로컬 IP 주소는 라우팅에 사용되며 문제 해결에도 도움이 됩니다. 예를 들어, 현재 기본 유닛에 항상 연결되어 있는 기본 클러스터 IP 주소에 연결하여 클러스터를 관리할 수 있습니다. 로컬 IP 주소에 연결하여 개별 멤버를 관리할 수 있습니다. TFTP 또는 syslog 같은 아웃바운드 관리 트래픽의 경우, 기본 유닛을 비롯한 각 유닛에서 로컬 IP 주소를 사용하여 서버에 연결합니다.

Firepower Threat Defense의 경우, 동일한 네트워크의 각 유닛에 관리 IP 주소를 할당합니다. 각 유닛을 Management Center에 추가할 때 이러한 IP 주소를 사용합니다.

Spanned EtherChannel

섀시당 하나 이상의 인터페이스를 클러스터 내의 모든 섀시를 포괄하는 EtherChannel로 그룹화할 수 있습니다. EtherChannel에서는 채널에서 사용 가능한 모든 활성 인터페이스 전반의 트래픽을 취합합니다. Spanned EtherChannel은 라우팅 및 투명 방화벽 모드에서 모두 구성할 수 있습니다. 라우팅 모드의 경우 EtherChannel은 단일 IP 주소를 통해 라우팅된 인터페이스로 구성됩니다. 투명 모드의 경우 IP 주소가 BVI에 할당되며 브리지 그룹 멤버 인터페이스에는 할당되지 않습니다. EtherChannel은 기본적인 작업 시 로드 밸런싱을 함께 제공합니다.



사이트 간 클러스터링

사이트 간 설치의 경우 권장 지침을 준수하여 클러스터링을 활용할 수 있습니다.

개별 사이트 ID에 속하는 각 클러스터 섀시를 구성할 수 있습니다.

사이트 ID는 사이트별 MAC 주소 및 IP 주소와 연동됩니다. 클러스터에서 가져온 패킷은 사이트별 MAC 주소 및 IP 주소를 사용하는 반면 클러스터에서 수신된 패킷은 글로벌 MAC 주소 및 IP 주소를 사용합니다. 이 기능은 스위치가 다른 두 개의 포트에서 두 개의 사이트의 동일한 글로벌 MAC 주소를 확인하는 것을 방지합니다. 대신 사이트 MAC 주소만 확인합니다. 사이트별 MAC 주소 및 IP 주소는 Spanned EtherChannel만을 사용하는 라우팅 모드에서 지원됩니다.

또한, 사이트 ID는 LISP 검사를 사용하여 플로우 모빌리티를 활성화하는 데 사용되며, 관리자 현지화는 성능을 개선하고 데이터 센터에 대한 사이트 간 클러스터링을 위해 왕복 시간 레이턴시를 줄이는 데 사용됩니다.

사이트 간 클러스터링에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- 데이터 센터 인터커넥트 크기 조정 —클러스터링의 사전 요구 사항, 158 페이지
- 사이트 간 지침 —클러스터링 지침, 160 페이지
- 사이트 간 예시 —사이트 간 클러스터링 예시, 177 페이지

클러스터링의 사전 요구 사항

섀시 간 하드웨어 및 소프트웨어 요건

클러스터의 모든 섀시는 다음과 같아야 합니다.

- Firepower 4100 Series: 모든 섀시는 동일한 모델이어야 합니다. Firepower 9300의 경우 모든 보안 모듈은 동일한 유형이어야 합니다. 섀시에 있는 모든 모듈은 빈 슬롯을 포함하여 클러스터에 속해야 하지만, 각 섀시에는 서로 다른 수량의 보안 모듈을 설치할 수 있습니다.
- 이미지 업그레이드 시 동일한 FXOS 소프트웨어 예외를 실행해야 합니다.
- 클러스터에 할당한 인터페이스와 동일한 인터페이스 컨피그레이션(예: 동일한 관리 인터페이스, EtherChannels, 활성화된 인터페이스, 속도 및 듀플렉스)을 포함해야 합니다. 동일한 인터페이스 ID에 대해 용량이 일치하고 인터페이스가 동일한 Spanned EtherChannel에서 성공적으로 번들링될 수 있는 한 섀시에서 서로 다른 네트워크 모듈 유형을 사용할 수 있습니다. 모든 데이터 인터페이스는 섀시 간 클러스터링에서 EtherChannel이어야 합니다.
- 동일한 NTP 서버를 사용해야 합니다. Firepower Threat Defense의 경우 Firepower Management Center 또한 동일한 NTP 서버를 사용해야 합니다. 시간을 수동으로 설정하지 마십시오.
- ASA의 경우 각 FXOS 섀시를 License Authority 또는 Satellite 서버에 등록해야 합니다. 보조 유닛에는 추가 비용이 없습니다. 영구 라이선스 예약의 경우, 각 섀시의 개별 라이선스를 구매해야 합니다. Firepower Threat Defense의 경우 모든 라이선싱이 Firepower Management Center에서 처리됩니다.

섀시 간 클러스터링을 위한 스위치 사전 요구 사항

- Firepower 4100/9300 섀시에서 클러스터링을 구성하기 전에 스위치 컨피그레이션을 완료하고 섀시의 모든 EtherChannel을 스위치에 성공적으로 연결합니다.
- 지원되는 스위치의 목록은 Cisco FXOS 호환성을 참조하십시오.

섀시 간 클러스터링을 위한 데이터 센터 인터커넥트 크기 조정

클러스터 제어 링크 트래픽용 DCI(data center interconnect) 대역폭은 다음 계산과 같이 예약해야 합니다.

$$\frac{\text{사이트당 클러스터 멤버의 수}}{2} \times \text{멤버당 클러스터 제어 링크 크기}$$

각 사이트의 멤버 수가 다를 경우, 더 큰 숫자를 계산에 사용합니다. DCI의 최소 대역폭은 한 멤버에 대한 클러스터 제어 링크의 크기보다 작으면 안 됩니다.

예를 들면 다음과 같습니다.

- 2개 사이트에 멤버가 4개인 경우:
  - 총 클러스터 멤버 4개
  - 사이트당 멤버 2개
  - 멤버당 5Gbps 클러스터 제어 링크

  예약된 DCI 대역폭 = 5Gbps(2/2 x 5Gbps)
- 3개 사이트에 멤버가 6개인 경우 크기가 다음과 같이 증가함:

- 총 클러스터 멤버 6개
- 사이트 1에 멤버 3개, 사이트 2에 멤버 2개, 사이트 3에 멤버 1개
- 멤버당 10Gbps 클러스터 제어 링크

예약된 DCI 대역폭 = 15Gbps(3/2 x 10Gbps)

- 2개 사이트에 멤버가 2개인 경우:
  - 총 클러스터 멤버 2개
  - 사이트당 멤버 1개
  - 멤버당 10Gbps 클러스터 제어 링크

예약된 DCI 대역폭 = 10Gbps(1/2 x 10Gbps = 5Gbps). 그러나 최소 대역폭은 클러스터 제어 링크의 크기(10Gbps)보다 작으면 안 됩니다.

클러스터링 지침

모델

- Firepower 9300의 ASA — 섀시 내, 섀시 간 및 사이트 간 클러스터링에 대해 지원됨
- Firepower 4100 Series의 ASA — 섀시 간 및 사이트 간 클러스터링에 대해 지원됨
- Firepower 9300의 Firepower Threat Defense — 섀시 내 및 섀시 간 클러스터링에 대해 지원됨
- Firepower 4100 Series의 Firepower Threat Defense — 섀시 간 클러스터링에 대해 지원됨
- Radware DefensePro — ASA와의 섀시 내 클러스터링에 대해 지원됨
- Radware DefensePro — Firepower Threat Defense와의 섀시 내 클러스터링에 대해 지원됨

섀시 간 클러스터링을 위한 스위치

- ASR 9006의 경우 기본이 아닌 MTU를 설정하려면 ASR 인터페이스 MTU를 클러스터 디바이스 MTU보다 14바이트 높게 설정합니다. 그렇지 않으면 **mtu-ignore** 옵션을 사용하지 않는 한 OSPF 인접 피어링 시도에 실패할 수 있습니다. 클러스터 디바이스 MTU는 ASR *IPv4* MTU와 일치해야 합니다.
- 클러스터 제어 링크 인터페이스용 스위치의 경우, 선택적으로 클러스터 유닛에 연결된 스위치 포트에서 Spanning Tree PortFast를 활성화하여 새 유닛에 대한 조인 프로세스 속도를 높일 수 있습니다.
- 스위치에서 Spanned EtherChannel의 번들링 속도가 저하될 경우, 스위치의 개별 인터페이스에 대한 LACP 속도를 빠르게 설정할 수 있습니다. Nexus Series 같은 일부 스위치는 ISSU(in-service software upgrade)를 수행할 때 빠른 LACP 속도를 지원하지 않습니다. 따라서 클러스터링과 ISSU를 함께 사용하지 않는 것이 좋습니다.

- 스위치에서는 **source-dest-ip** 또는 **source-dest-ip-port** EtherChannel 로드 밸런싱 알고리즘 중 하나를 사용하는 것이 좋습니다(Cisco Nexus OS 및 Cisco IOS **port-channel load-balance** 명령 참조). 로드 밸런싱 알고리즘에서는 **vlan** 키워드를 사용하지 마십시오. 이렇게 할 경우 클러스터의 디바이스에 트래픽이 균일하지 않게 분산될 수 있기 때문입니다.
- 스위치에서 EtherChannel의 로드 밸런싱 알고리즘을 변경할 경우, 스위치의 EtherChannel 인터페이스에서 트래픽 전달이 일시적으로 중단되며 Spanning Tree Protocol이 재시작됩니다. 트래픽에서 플로우를 다시 시작하기 전까지 지연이 발생하게 됩니다.
- 클러스터 제어 링크 경로의 스위치에서는 L4 체크섬을 확인하지 않습니다. 클러스터 제어 링크를 통해 리디렉션된 트래픽에는 올바른 L4 체크섬이 없습니다. L4 체크섬을 확인하는 스위치의 경우 트래픽이 감소하는 결과를 초래할 수 있습니다.
- 포트 채널 번들링 다운타임은 구성된 keepalive 기간을 초과하면 안 됩니다.
- Supervisor 2T EtherChannel에서 기본 해시 분산 알고리즘은 적응형입니다. VSS 설계에서 비대칭 트래픽을 방지하려면 클러스터 디바이스에 연결된 포트 채널의 해시 알고리즘을 다음과 같이 변경하여 수정합니다.

  router(config)# **port-channel** *id***hash-distributionfixed**

  VSS 피어 링크의 적응형 알고리즘을 활용할 때가 있을 수 있으므로 알고리즘을 전체적으로 변경하지 마십시오.

섀시 간 클러스터링을 위한 **EtherChannel**

- 연결 스위치의 경우, EtherChannel 모드를 Active(활성)로 설정합니다. On(켜짐) 모드는 Firepower 4100/9300 섀시에서 지원되지 않으며 클러스터 제어 링크에서도 지원되지 않습니다.
- FXOS EtherChannel은 기본적으로 LACP 속도가 보통으로 설정되어 있습니다. 이렇게 설정하면 포트-채널 멤버를 번들링하는 데 30초 넘게 걸릴 수 있으므로 이로 인해 클러스터 인터페이스 상태 확인에 장애가 발생하여 클러스터에서 유닛이 제거될 수 있습니다. 따라서 LACP 속도를 빠르게 변경하는 것이 좋습니다. 다음 예에서는 "기본" LACP 정책을 다음과 같이 수정합니다.

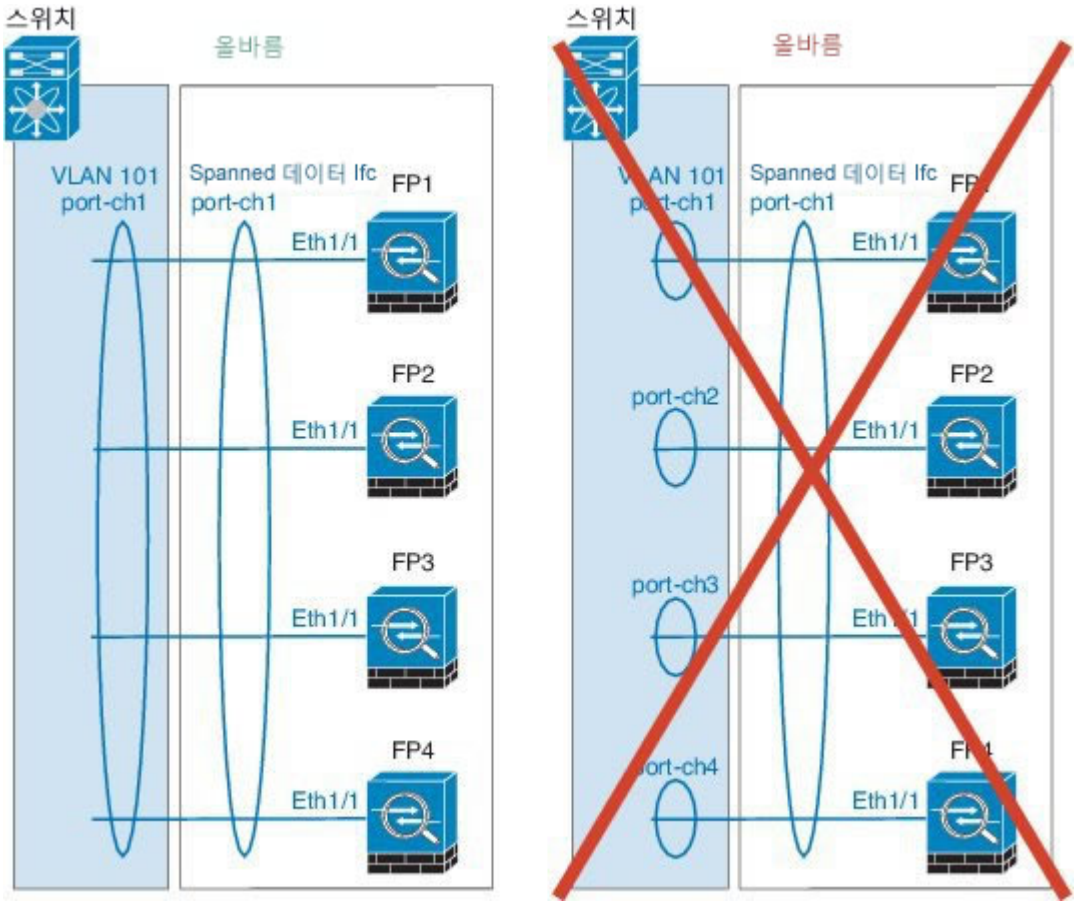
```
firepower# scope org
firepower /org # scope lacppolicy default
firepower /org/lacppolicy# set lacp-rate fast
firepower /org* # commit-buffer
```

참고 Nexus Series 같은 일부 스위치는 ISSU(in-service software upgrade, 서비스 중 소프트웨어 업그레이드)를 수행할 때 빠른 LACP 속도를 지원하지 않습니다. 따라서 클러스터링과 ISSU를 함께 사용하지 않는 것이 좋습니다.

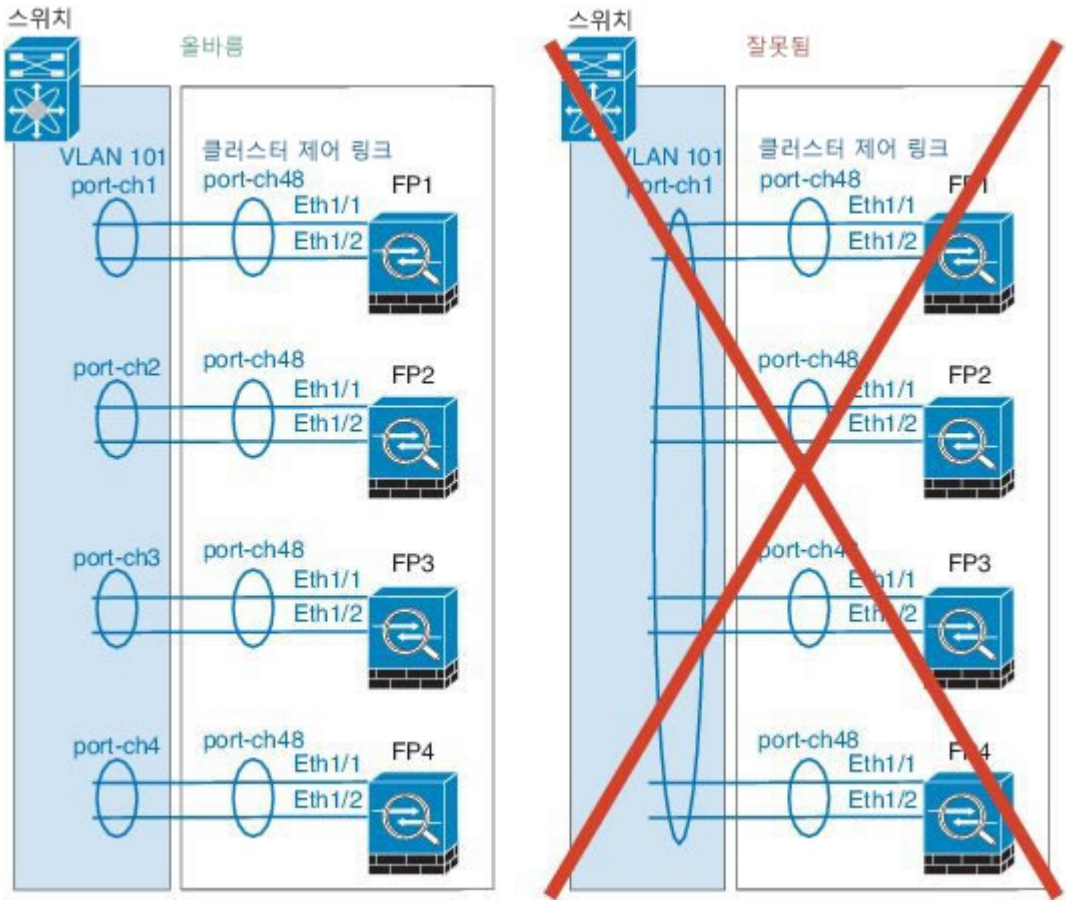
- 15.1(1)S2 이전 Catalyst 3750-X Cisco IOS 소프트웨어 버전에서는 클러스터 유닛에서 EtherChannel과 스위치 스택 간 연결을 지원하지 않았습니다. 기본 스위치 설정으로 클러스터 유닛 EtherChannel이 교차 스택에 연결되어 있는 상태에서 마스터 스위치의 전원이 꺼질 경우, 나머지 스위치에 연결된 EtherChannel은 작동하지 않습니다. 호환성을 개선하려면 **stack-mac persistent timer** 명

령을 다시 로드할 수 있을 정도로 충분히 큰 값으로 설정합니다(예: 8분 또는 무한인 경우 0). 또는 15.1(1)S2 같은 더 안정적인 스위치 소프트웨어 버전으로 업그레이드할 수 있습니다.

- Spanned EtherChannel 컨피그레이션과 디바이스-로컬 EtherChannel 컨피그레이션 — Spanned EtherChannel와 디바이스-로컬 EtherChannel에서 각각 알맞게 스위치를 구성해야 합니다.
  - Spanned EtherChannel — 클러스터의 모든 멤버 전체를 포괄하는 클러스터 유닛 *Spanned* EtherChannels의 경우, 인터페이스가 스위치의 단일 EtherChannel에 통합됩니다. 각 인터페이스가 스위치의 동일한 채널 그룹에 있는지 확인하십시오.



  - 디바이스-로컬 EtherChannel — 클러스터 제어 링크에 대해 구성된 모든 EtherChannel을 비롯한 클러스터 유닛 디바이스-로컬 EtherChannel의 경우, 스위치에서 별도의 EtherChannel을 구성해야 합니다. 여러 클러스터 유닛 EtherChannel을 스위치에서 하나의 EtherChannel에 통합하지 마십시오.



추가 지침

- 이중화를 위해 EtherChannel을 VSS 또는 vPC에 연결하는 것이 좋습니다.
- 섀시 내에서 일부 보안 모듈을 클러스터링하여 독립형 모드에서 다른 보안 모듈을 실행할 수 없습니다. 클러스터에 모든 보안 모듈을 포함해야 합니다.

클러스터링 기본값

클러스터 제어 링크는 포트 채널 48을 사용합니다.

ASA 클러스터링 구성

Firepower 4100/9300 섀시 수퍼바이저에서 손쉽게 클러스터를 구축할 수 있습니다. 모든 초기 컨피그레이션은 유닛마다 자동으로 생성됩니다. 섀시 간 클러스터링의 경우 각 섀시를 개별적으로 구성해야 합니다. 한 섀시에 클러스터를 구축한 다음 손쉬운 구축을 위해 첫 번째 섀시의 부트스트랩 컨피그레이션을 다음 섀시에 복사합니다.

시작하기 전에

- 모듈을 설치하지 않은 경우에도 Firepower 9300 섀시에서 3개의 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개의 모듈을 모두 구성하지 않은 경우, 클러스터가 나타나지 않습니다.
- **Interfaces(**인터페이스**)** 탭에서 포트 채널 48 클러스터 유형 인터페이스는 멤버 인터페이스를 포함하지 않은 경우 **Operation State(**작동 상태**)**를 **failed(**실패함**)**로 표시합니다. 섀시 내 클러스터링의 경우 이 EtherChannel이 멤버 인터페이스를 필요로 하지 않으므로 이 작동 상태를 무시할 수 있습니다.
- Firepower 4100/9300 섀시에서 라우팅된 방화벽 모드 ASA 클러스터만 구축할 수 있습니다.

절차

단계 **1** 클러스터를 구축하기 전에 데이터 유형 인터페이스 또는 EtherChannel(포트 채널)을 하나 이상 구성합니다. 포트 채널 생성, 142 페이지 또는 인터페이스 속성 편집, 141 페이지를 참조하십시오.
모든 인터페이스는 클러스터에 기본적으로 할당되어 있습니다. 또한 데이터 인터페이스를 구축한 후에 클러스터에 추가할 수 있습니다.

섀시 간 클러스터링의 경우, 모든 데이터 인터페이스는 멤버 인터페이스가 최소 1개 있는 EtherChannel이어야 합니다. 각 섀시에서 EtherChannel을 추가합니다.

단계 **2** 관리 유형 인터페이스 또는 EtherChannel을 추가합니다. 포트 채널 생성, 142 페이지 또는 인터페이스 속성 편집, 141 페이지를 참조하십시오.

단계 **3** 포트 채널 48은 클러스터 제어 링크로 예약됩니다. 섀시 간 클러스터링의 경우 최소 1개의 멤버 인터페이스를 포트 채널 48에 추가합니다.

단계 **4** 보안 서비스 모드를 시작합니다.
scopessa

예제:

```
Firepower # scope ssa
Firepower /ssa #
```

단계 **5** 클러스터를 생성합니다.
enter logical-device *device_name***asa "1,2,3" clustered**

예제:

```
Firepower /ssa # enter logical-device ASA1 asa "1,2,3" clustered
Firepower /ssa/logical-device* #
```

*device_name*은 Firepower 4100/9300 섀시 수퍼바이저가 클러스터링 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 클러스터 이름이 아닙니다. 하드웨어를 아직 설치하지 않은 경우에도 보안 모듈 3개를 모두 지정해야 합니다.

참고 모듈을 설치하지 않은 경우에도 섀시에서 3개의 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개의 모듈을 모두 구성하지 않은 경우, 클러스터가 나타나지 않습니다.

단계 **6** 클러스터 파라미터를 구성합니다.
enter cluster-bootstrap

예제:

```
Firepower /ssa/logical-device* # enter cluster-bootstrap
Firepower /ssa/logical-device/cluster-bootstrap* #
```

단계 **7** 보안 모듈 컨피그레이션에서 클러스터 그룹 이름을 설정합니다.
set service-type *cluster_name*

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set service-type cluster1
Firepower /ssa/logical-device/cluster-bootstrap* #
```

이름은 1~38자로 된 ASCII 문자열이어야 합니다.

단계 **8** 클러스터 인터페이스 모드를 설정합니다.
set mode spanned-etherchannel

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set mode spanned-etherchannel
Firepower /ssa/logical-device/cluster-bootstrap* #
```

Spanned EtherChannel 모드는 유일하게 지원되는 모드입니다.

단계 **9** 관리 IP 주소 정보를 구성합니다.
이 정보는 보안 모듈 컨피그레이션의 관리 인터페이스를 구성하는 데 사용됩니다.

a) 로컬 IP 주소의 풀을 구성합니다. 이 중 하나는 인터페이스의 각 클러스터 유닛에 할당됩니다.
set ipv4 pool *start_ip end_ip*

set ipv6 pool *start_ip end_ip*

최소한 클러스터에 있는 유닛 수에 상응하는 개수의 주소를 포함해야 합니다. Firepower 9300의 경우, 모든 모듈 슬롯을 채우지 않은 경우에도 섀시당 3개의 주소를 포함해야 합니다. 클러스터를 확장하려는 경우, 추가 주소를 포함하십시오. 현재 기본 유닛에 속하는 가상 IP 주소(기본 클러스터 IP 주소)는 이러한 풀에 속하지 않습니다. 따라서 동일한 네트워크에서 기본 클러스터 IP 주소에 대한 IP 주소를 예약해 두어야 합니다. IPv4 및/또는 IPv6 주소를 사용할 수 있습니다.

b) 관리 인터페이스의 기본 클러스터 IP 주소를 구성합니다.
set virtual ipv4 *ip_address***mask** *mask*

set virtual ipv6 *ip_address***prefix-length** *prefix*

이 IP 주소는 같은 네트워크의 클러스터 풀 주소로 있어야 하지만 풀의 일부는 아닙니다.

c) 네트워크 게이트웨이 주소를 입력합니다.
set ipv4 gateway *ip_address*

set ipv6 gateway *ip_address*

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv4 gateway 10.1.1.254
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv4 pool 10.1.1.11 10.1.1.27
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv6 gateway 2001:DB8::AA
Firepower /ssa/logical-device/cluster-bootstrap* # set ipv6 pool 2001:DB8::11 2001:DB8::27
Firepower /ssa/logical-device/cluster-bootstrap* # set virtual ipv4 10.1.1.1 mask
255.255.255.0
Firepower /ssa/logical-device/cluster-bootstrap* # set virtual ipv6 2001:DB8::1 prefix-length
 64
```

단계 **10** 섀시 ID를 설정합니다.
set chassis-id *id*

클러스터의 각 섀시에는 고유한 ID가 필요합니다.

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set chassis-id 1
Firepower /ssa/logical-device/cluster-bootstrap* #
```

단계 **11** 사이트 간 클러스터링의 경우 사이트 ID를 1~8로 설정합니다.
set site-id *number*

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set site-id 1
Firepower /ssa/logical-device/cluster-bootstrap* #
```

단계 **12** 클러스터 제어 링크의 제어 트래픽에 대해 인증 키를 구성합니다.
set key

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set key
Key: diamonddogs
Firepower /ssa/logical-device/cluster-bootstrap* #
```

공유 암호를 입력하라는 메시지가 표시됩니다.

공유 암호는 1~63자로 된 ASCII 문자열입니다. 공유 암호는 키를 생성하는 데 사용됩니다. 이 옵션은 연결 상태 업데이트 및 전달된 패킷을 비롯한 데이터 경로 트래픽에 영향을 미치지 않으며, 항상 일반 텍스트로 전송됩니다.

단계 **13** 클러스터 부트스트랩 모드 및 논리적 디바이스 모드를 종료합니다.
exit

exit

단계 **14** 사용 가능한 소프트웨어 버전을 확인한 다음 사용할 버전을 설정합니다.

a) 사용 가능한 버전을 표시합니다.
show app

예제:

```
/ssa # show app

Application:
    Name       Version    Description Author     Deploy Type  CSP Type    Is Default App

    ---------- ---------- ----------- ---------- -----------  ----------- --------------

    asa        9.1.4.152  N/A         cisco      Native       Application Yes
    asa        9.4.2      N/A         cisco      Native       Application No
    asa        9.5.2.1    N/A         cisco      Native       Application No
```

b) 사용할 버전의 앱 모드를 시작합니다.
scope app asa *version_number*

c) 이 버전을 기본값으로 설정합니다.
set-default

d) 앱 모드를 종료합니다.
exit

예제:

```
/ssa* # scope app asa 9.5.2.1
/ssa/app* # set-default
/ssa/app* # exit
/ssa* #
```

단계 **15** 컨피그레이션을 커밋합니다.
commit-buffer

Firepower 4100/9300 섀시 수퍼바이저는 기본 보안 모듈 소프트웨어 버전을 다운로드하고 클러스터 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 각 보안 모듈에 입력하여 클러스터를 구축합니다.

단계 **16** 클러스터에 다른 섀시를 추가하려면 고유한 **chassis-id**및 올바른 **site-id**를 구성해야 하는 경우를 제외하고 이 절차를 반복합니다. 아니면 두 섀시 모두에 동일한 컨피그레이션을 사용합니다.

단계 **17** 클러스터링 컨피그레이션을 맞춤 설정하려면 기본 유닛 보안 모듈에 연결합니다.

예

섀시 1의 경우:

```
scope eth-uplink
  scope fabric a
    enter port-channel 1
      set port-type data
      enable
      enter member-port Ethernet1/1
        exit
      enter member-port Ethernet1/2
        exit
      exit
    enter port-channel 2
```

```
      set port-type data
      enable
      enter member-port Ethernet1/3
        exit
      enter member-port Ethernet1/4
        exit
      exit
    enter port-channel 3
      set port-type data
      enable
      enter member-port Ethernet1/5
        exit
      enter member-port Ethernet1/6
        exit
      exit
    enter port-channel 4
      set port-type mgmt
      enable
      enter member-port Ethernet2/1
        exit
      enter member-port Ethernet2/2
        exit
      exit
    enter port-channel 48
      set port-type cluster
      enable
      enter member-port Ethernet2/3
        exit
      exit
    exit
  exit
commit-buffer

scope ssa
  enter logical-device ASA1 asa "1,2,3" clustered
    enter cluster-bootstrap
      set chassis-id 1
      set ipv4 gateway 10.1.1.254
      set ipv4 pool 10.1.1.11 10.1.1.27
      set ipv6 gateway 2001:DB8::AA
      set ipv6 pool 2001:DB8::11 2001:DB8::27
      set key
      Key: f@arscape
      set mode spanned-etherchannel
      set service-type cluster1
      set virtual ipv4 10.1.1.1 mask 255.255.255.0
      set virtual ipv6 2001:DB8::1 prefix-length 64
      exit
    exit
  scope app asa 9.5.2.1
    set-default
    exit
  commit-buffer
```

섀시 2의 경우:

```
scope eth-uplink
  scope fabric a
    create port-channel 1
      set port-type data
      enable
      create member-port Ethernet1/1
        exit
      create member-port Ethernet1/2
        exit
      exit
    create port-channel 2
      set port-type data
      enable
      create member-port Ethernet1/3
        exit
      create member-port Ethernet1/4
```

```
        exit
      exit
    create port-channel 3
      set port-type data
      enable
      create member-port Ethernet1/5
        exit
      create member-port Ethernet1/6
        exit
      exit
    create port-channel 4
      set port-type mgmt
      enable
      create member-port Ethernet2/1
        exit
      create member-port Ethernet2/2
        exit
      exit
    create port-channel 48
      set port-type cluster
      enable
      create member-port Ethernet2/3
        exit
      exit
    exit
  exit
commit-buffer

scope ssa
  enter logical-device ASA1 asa "1,2,3" clustered
    enter cluster-bootstrap
      set chassis-id 2
      set ipv4 gateway 10.1.1.254
      set ipv4 pool 10.1.1.11 10.1.1.15
      set ipv6 gateway 2001:DB8::AA
      set ipv6 pool 2001:DB8::11 2001:DB8::19
      set key
      Key: f@rscape
      set mode spanned-etherchannel
      set service-type cluster1
      set virtual ipv4 10.1.1.1 mask 255.255.255.0
      set virtual ipv6 2001:DB8::1 prefix-length 64
      exit
    exit
  scope app asa 9.5.2.1
    set-default
    exit
  commit-buffer
```

Firepower Threat Defense 클러스터링 구성

Firepower 4100/9300 섀시 수퍼바이저에서 손쉽게 클러스터를 구축할 수 있습니다. 모든 초기 컨피그레이션은 유닛마다 자동으로 생성됩니다. 섀시 간 클러스터링의 경우 각 섀시를 개별적으로 구성해야 합니다. 한 섀시에 클러스터를 구축한 다음 손쉬운 구축을 위해 첫 번째 섀시의 부트스트랩 컨피그레이션을 다음 섀시에 복사합니다.

시작하기 전에

- 모듈을 설치하지 않은 경우에도 Firepower 9300 섀시에서 3개의 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개의 모듈을 모두 구성하지 않은 경우, 클러스터가 나타나지 않습니다.

- **Interfaces(**인터페이스**)** 탭에서 포트 채널 48 클러스터 유형 인터페이스는 멤버 인터페이스를 포함하지 않은 경우 **Operation State(**작동 상태**)**를 **failed(**실패함**)**로 표시합니다. 섀시 내 클러스터링의 경우 이 EtherChannel이 멤버 인터페이스를 필요로 하지 않으므로 이 작동 상태를 무시할 수 있습니다.

절차

단계 **1** 클러스터를 구축하기 전에 데이터 유형 인터페이스 또는 EtherChannel(포트 채널)을 하나 이상 구성합니다. 포트 채널 생성, 142 페이지 또는 인터페이스 속성 편집, 141 페이지를 참조하십시오.
또한 데이터 인터페이스를 구축한 후에 클러스터에 추가할 수 있습니다.

섀시 간 클러스터링의 경우, 모든 데이터 인터페이스는 멤버 인터페이스가 최소 1개 있는 EtherChannel이어야 합니다. 각 섀시에서 EtherChannel을 추가합니다.

단계 **2** (선택 사항) 클러스터를 구축하기 전에 Firepower 이벤트 유형 인터페이스를 구성합니다. 인터페이스 속성 편집, 141 페이지를 참조하십시오.
이 인터페이스는 Firepower Threat Defense 디바이스의 보조 관리 인터페이스입니다. 이 인터페이스를 사용하려면 Firepower Threat Defense CLI에서 해당 IP 주소 및 기타 파라미터를 구성해야 합니다. 예를 들면 관리 트래픽을 이벤트(예: 웹 이벤트)에서 분리할 수 있습니다. Firepower Management Center 명령 참조에서 **configure network** 명령을 참조하십시오.

단계 **3** 관리 유형 인터페이스 또는 EtherChannel을 추가합니다. 포트 채널 생성, 142 페이지 또는 인터페이스 속성 편집, 141 페이지를 참조하십시오.

단계 **4** 포트 채널 48은 클러스터 제어 링크로 예약됩니다. 섀시 간 클러스터링의 경우 최소 1개의 멤버 인터페이스를 포트 채널 48에 추가합니다.

단계 **5** 보안 서비스 모드를 시작합니다.
scopessa

예제:

```
Firepower # scope ssa
Firepower /ssa #
```

단계 **6** 클러스터를 생성합니다.
enter logical-device *device_name***ftd "1,2,3" clustered**

예제:

```
Firepower /ssa # enter logical-device FTD1 ftd "1,2,3" clustered
Firepower /ssa/logical-device* #
```

*device_name*은 Firepower 4100/9300 섀시 수퍼바이저가 클러스터링 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 클러스터 이름이 아닙니다.

참고 모듈을 설치하지 않은 경우에도 섀시에서 3개의 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개의 모듈을 모두 구성하지 않은 경우, 클러스터가 나타나지 않습니다.

단계 **7** 클러스터 부트스트랩 파라미터를 구성합니다.

a) 클러스터 부트스트랩 객체를 생성합니다.
enter cluster-bootstrap

b) 섀시 ID를 설정합니다.

set chassis-id *id*

클러스터의 각 섀시에는 고유한 ID가 필요합니다.

c) 보안 모듈 컨피그레이션에서 클러스터 키를 설정합니다.

set key

공유 암호를 입력하라는 메시지가 표시됩니다.

공유 암호는 1~63자로 된 ASCII 문자열입니다. 공유 암호는 키를 생성하는 데 사용됩니다. 이 옵션은 연결 상태 업데이트 및 전달된 패킷을 비롯한 데이터 경로 트래픽에 영향을 미치지 않으며, 항상 일반 텍스트로 전송됩니다.

d) 클러스터 인터페이스 모드를 설정합니다.

set mode spanned-etherchannel

Spanned EtherChannel 모드는 유일하게 지원되는 모드입니다.

e) 보안 모듈 컨피그레이션에서 클러스터 그룹 이름을 설정합니다.

set service-type *cluster_name*

이름은 1~38자로 된 ASCII 문자열이어야 합니다.

f) 클러스터 부트스트랩 모드를 종료합니다.
exit

예제:

```
Firepower /ssa/logical-device* # enter cluster-bootstrap
Firepower /ssa/logical-device/cluster-bootstrap* # set chassis-id 1
Firepower /ssa/logical-device/cluster-bootstrap* # set key
  Key: f@arscape
Firepower /ssa/logical-device/cluster-bootstrap* # set mode spanned-etherchannel
Firepower /ssa/logical-device/cluster-bootstrap* # set service-type cluster1
Firepower /ssa/logical-device/cluster-bootstrap* # exit
Firepower /ssa/logical-device/* #
```

단계 **8** 관리 부트스트랩 파라미터를 구성합니다.

a) 관리 부트스트랩 객체를 생성합니다.
entermgmt-bootstrapftd

b) 관리하는 Firepower Management Center의 IP 주소를 지정합니다.
enterbootstrap-keyFIREPOWER_MANAGER_IP

setvalue *IP_address*

exit

c) 논리적 디바이스가 작동할 모드(Routed(라우팅됨) 또는 Transparent(투명))를 지정합니다.
enterbootstrap-keyFIREWALL_MODE

setvalue *firewall_mode*

exit

d) 디바이스와 Firepower Management Center 간에 공유할 키를 지정합니다.
enterbootstrap-key-secretREGISTRATION_KEY

setvalue

registration_key

exit

e) 논리적 디바이스에 사용할 비밀번호를 지정합니다.
enterbootstrap-key-secretPASSWORD

setvalue

password

exit

f) 논리적 디바이스의 정규화된 호스트 이름을 지정합니다.
enterbootstrap-keyFQDN

setvalue *fqdn*

exit

g) 논리적 디바이스에서 사용할 쉼표로 구분된 DNS 서버 목록을 지정합니다.
enterbootstrap-keyDNS_SERVERS

setvalue *dns_servers*

exit

h) 논리적 디바이스를 위한 검색 도메인의 쉼표로 구분된 목록을 지정합니다.
enterbootstrap-keySEARCH_DOMAINS

setvalue *search_domains*

exit

i) 클러스터의 각 보안 모듈에 대해 관리 IP 주소를 구성합니다.

참고 Firepower 9300의 경우, 모듈을 설치하지 않은 경우에도 섀시에 있는 3개의 모듈 슬롯 모두에 대해 IP 주소를 설정해야 합니다. 3개의 모듈을 모두 구성하지 않은 경우, 클러스터가 나타나지 않습니다.

다음과 같이 IPv4 관리 인터페이스 객체를 생성합니다.

1 관리 인터페이스 객체를 생성합니다.

enteripv4 *slot_id***firepower**

2 게이트웨이 주소를 설정합니다.

setgateway *gateway_address*

3 IP 주소 및 마스크를 설정합니다.

setip *ip_address***mask** *network_mask*

4 관리 IP 모드를 종료합니다.

exit

5 섀시에 있는 나머지 모듈에 대해 반복합니다.

다음과 같이 IPv6 관리 인터페이스 객체를 생성합니다.

1. 관리 인터페이스 객체를 생성합니다.
   enteripv6 *slot_id***firepower**
2. 게이트웨이 주소를 설정합니다.
   setgateway *gateway_address*
3. IP 주소 및 접두사를 설정합니다.
   set ip *ip_address***prefix-length** *prefix*
4. 관리 IP 모드를 종료합니다.
   exit
5. 섀시에 있는 나머지 모듈에 대해 반복합니다.

j) 관리 부트스트랩 모드를 종료합니다.
exit

예제:

```
Firepower /ssa/logical-device* # enter mgmt-bootstrap ftd
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key FIREPOWER_MANAGER_IP
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 10.0.0.100
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key FIREWALL_MODE
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value routed
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key-secret REGISTRATION_KEY
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
  Value: ziggy$tardust
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
  Value: $pidersfrommars
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key FQDN
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value example.cisco.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key DNS_SERVERS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 192.168.1.1
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key SEARCH_DOMAINS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value example.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter ipv4 1 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.31 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter ipv4 2 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.32 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter ipv4 3 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.33 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* #
```

단계 **9** 논리적 디바이스 모드를 종료합니다.

exit

단계 **10** 사용 가능한 소프트웨어 버전을 확인한 다음 사용할 버전을 설정합니다.

a) 사용 가능한 버전을 표시합니다.
show app

예제:

```
/ssa # show app

Application:
    Name       Version    Description Author     Deploy Type CSP Type    Is Default App
    ---------- ---------- ----------- ---------- ----------- ----------- --------------
    ftd        6.0.1.37   N/A         cisco      Native      Application Yes
    ftd        6.1.0.11   N/A         cisco      Native      Application No
    ftd        6.1.0.21   N/A         cisco      Native      Application No
```

b) 사용할 버전의 앱 모드를 시작합니다.
scope app ftd *version_number*

c) 이 버전을 기본값으로 설정합니다.
set-default

d) 이 버전의 엔드 유저 라이선스 계약에 동의합니다.
accept-license-agreement

e) 앱 모드를 종료합니다.
exit

예제:

```
/ssa # scope app ftd 6.1.0.21
/ssa/app # set-default
/ssa/app* # accept-license-agreement
/ssa/app* # exit
/ssa* #
```

단계 **11** 컨피그레이션을 커밋합니다.
commit-buffer

Firepower 4100/9300 섀시 수퍼바이저는 기본 보안 모듈 소프트웨어 버전을 다운로드하고 클러스터 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 각 보안 모듈에 입력하여 클러스터를 구축합니다.

단계 **12** 클러스터에 다른 섀시를 추가하려면 고유한 **chassis-id** 및 고유한 관리 IP 주소를 구성해야 하는 경우를 제외하고 이 절차를 반복합니다. 아니면 두 섀시 모두에 동일한 컨피그레이션을 사용합니다.

단계 **13** 관리 IP 주소를 사용하여 각 보안 모듈을 Firepower Management Center에 추가한 다음 웹 인터페이스에서 클러스터로 그룹화합니다.
모든 클러스터 유닛은 Firepower Management Center에 추가하기 전에 FXOS에서 성공적으로 구성된 클러스터에 있어야 합니다.

예

```
scope eth-uplink
  scope fabric a
    enter port-channel 1
      set port-type data
      enable
      create member-port Ethernet1/1
        exit
      create member-port Ethernet1/2
        exit
      exit
    enter port-channel 2
      set port-type data
      enable
      create member-port Ethernet1/3
        exit
      create member-port Ethernet1/4
        exit
      exit
    enter port-channel 3
      set port-type firepower-eventing
      enable
      create member-port Ethernet1/5
        exit
      create member-port Ethernet1/6
        exit
      exit
    enter port-channel 4
      set port-type mgmt
      enable
      create member-port Ethernet2/1
        exit
      enter member-port Ethernet2/2
        exit
      exit
    enter port-channel 48
      set port-type cluster
      enable
      enter member-port Ethernet2/3
        exit
      exit
    exit
  exit
commit-buffer

scope ssa
  enter logical-device FTD1 ftd "1,2,3" clustered
    enter cluster-bootstrap
      set chassis-id 1
      set key cluster_key
      set mode spanned-etherchannel
      set service-type ftd-cluster
      exit
    enter mgmt-bootstrap ftd
      enter bootstrap-key FIREPOWER_MANAGER_IP
        set value 10.0.0.100
        exit
      enter bootstrap-key FIREWALL_MODE
        set value transparent
        exit
      enter bootstrap-key-secret REGISTRATION_KEY
        set value
          Value: alladinsane
        exit
      enter bootstrap-key-secret PASSWORD
        set value
          Value: widthofacircle
        exit
      enter bootstrap-key FQDN
        set value ftd.cisco.com
        exit
```

```
      enter bootstrap-key DNS_SERVERS
        set value 192.168.1.1
        exit
      enter bootstrap-key SEARCH_DOMAINS
        set value search.com
        exit
      enter ipv4 1 firepower
        set gateway 10.0.0.1
        set ip 10.0.0.31 mask 255.255.255.0
        exit
      enter ipv4 2 firepower
        set gateway 10.0.0.1
        set ip 10.0.0.32 mask 255.255.255.0
        exit
      enter ipv4 3 firepower
        set gateway 10.0.0.1
        set ip 10.0.0.33 mask 255.255.255.0
        exit
      exit
    exit
  scope app ftd 6.0.0.837
    accept-license-agreement
    exit
  commit-buffer
```

섀시 2의 경우:

```
scope eth-uplink
  scope fabric a
    enter port-channel 1
      set port-type data
      enable
      create member-port Ethernet1/1
        exit
      create member-port Ethernet1/2
        exit
      exit
    enter port-channel 2
      set port-type data
      enable
      create member-port Ethernet1/3
        exit
      create member-port Ethernet1/4
        exit
      exit
    enter port-channel 3
      set port-type firepower-eventing
      enable
      create member-port Ethernet1/5
        exit
      create member-port Ethernet1/6
        exit
      exit
    enter port-channel 4
      set port-type mgmt
      enable
      create member-port Ethernet2/1
        exit
      enter member-port Ethernet2/2
        exit
      exit
    enter port-channel 48
      set port-type cluster
      enable
      enter member-port Ethernet2/3
        exit
      exit
    exit
  exit
commit-buffer

scope ssa
```

```
enter logical-device FTD1 ftd "1,2,3" clustered
  enter cluster-bootstrap
    set chassis-id 2
    set key cluster_key
    set mode spanned-etherchannel
    set service-type ftd-cluster
    exit
  enter mgmt-bootstrap ftd
    enter bootstrap-key FIREPOWER_MANAGER_IP
      set value 10.0.0.100
      exit
    enter bootstrap-key FIREWALL_MODE
      set value transparent
      exit
    enter bootstrap-key-secret REGISTRATION_KEY
      set value
        Value: alladinsane
      exit
    enter bootstrap-key-secret PASSWORD
      set value
        Value: widthofacircle
      exit
    enter bootstrap-key FQDN
      set value ftd.cisco.com
      exit
    enter bootstrap-key DNS_SERVERS
      set value 192.168.1.1
      exit
    enter bootstrap-key SEARCH_DOMAINS
      set value search.com
      exit
    enter ipv4 1 firepower
      set gateway 10.0.0.1
      set ip 10.0.0.31 mask 255.255.255.0
      exit
    enter ipv4 2 firepower
      set gateway 10.0.0.1
      set ip 10.0.0.32 mask 255.255.255.0
      exit
    enter ipv4 3 firepower
      set gateway 10.0.0.1
      set ip 10.0.0.33 mask 255.255.255.0
      exit
    exit
  exit
scope app ftd 6.0.0.837
  accept-license-agreement
  exit
commit-buffer
```

사이트 간 클러스터링 예시

다음 예에는 지원되는 클러스터 구축에 대한 내용이 나와 있습니다.

Spanned EtherChannel 투명 모드 North-South 사이트 간 예시

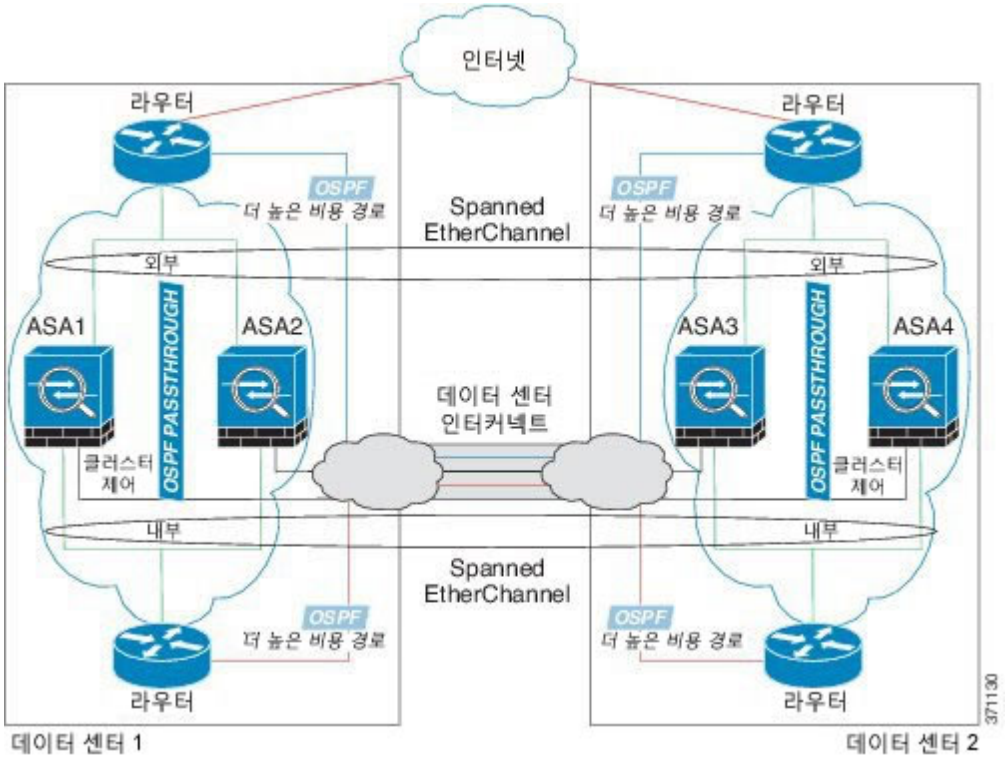
다음 예에서는 내부 라우터와 외부 라우터의 사이에 위치(North-South 삽입)한 2개 데이터 센터 각각에 2개의 클러스터 멤버가 있습니다. 클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부용 Spanned EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 섀시에 있습니다.

각 데이터 센터의 내부 및 외부 라우터는 투명 ASA를 통과하는 OSPF를 사용합니다. MAC과 달리 라우터 IP는 모든 라우터에서 고유합니다. DCI 전반에 비용이 높은 경로를 할당하면 특정 사이트의 모

든 클러스터 멤버가 다운되지 않는 한 각 데이터 센터 내에서 트래픽이 유지됩니다. ASA를 통과하는 비용이 낮은 경로의 경우, 클러스터의 각 사이트에 있는 동일한 브리지 그룹을 거쳐 비대칭 연결을 유지해야 합니다. 어떤 사이트의 모든 클러스터 멤버에 장애가 발생하는 경우, 각 라우터의 트래픽은 DCI를 통해 다른 사이트의 클러스터 멤버로 이동합니다.

각 사이트의 스위치 구현 과정에는 다음 사항이 포함될 수 있습니다.

- 사이트 간 VSS/vPC — 이 시나리오의 경우 데이터 센터 1에 하나의 스위치를 설치하고, 나머지 하나는 데이터 센터 2에 설치합니다. 각 데이터 센터의 클러스터 유닛에 사용할 수 있는 한 가지 옵션은 로컬 스위치에만 연결하는 한편, VSS/vPC 트래픽이 DCI를 통과하도록 하는 것입니다. 이 경우 연결의 대부분은 각 데이터 센터에 로컬로 저장됩니다. DCI에서 추가 트래픽을 처리할 수 있는 경우, 선택적으로 DCI 전반의 두 스위치에 각 유닛을 연결할 수 있습니다. 이 경우 트래픽이 데이터 센터 전반에 분산되므로 DCI의 성능이 매우 뛰어나야 합니다.
- 각 사이트의 로컬 VSS/vPC — 스위치 이중화를 개선하기 위해 각 사이트에 별도의 VSS/vPC 쌍을 2개씩 설치할 수 있습니다. 이 경우 여전히 클러스터 유닛의 Spanned EtherChannel은 두 로컬 스위치에만 연결된 데이터 센터 1 섀시 및 이 로컬 스위치에 연결된 데이터 센터 2 섀시로 이루어져 있으나, 사실상 Spanned EtherChannel은 "분리"되어 있습니다. 각 로컬 VSS/vPC에서는 Spanned EtherChannel을 사이트-로컬 EtherChannel로 간주합니다.

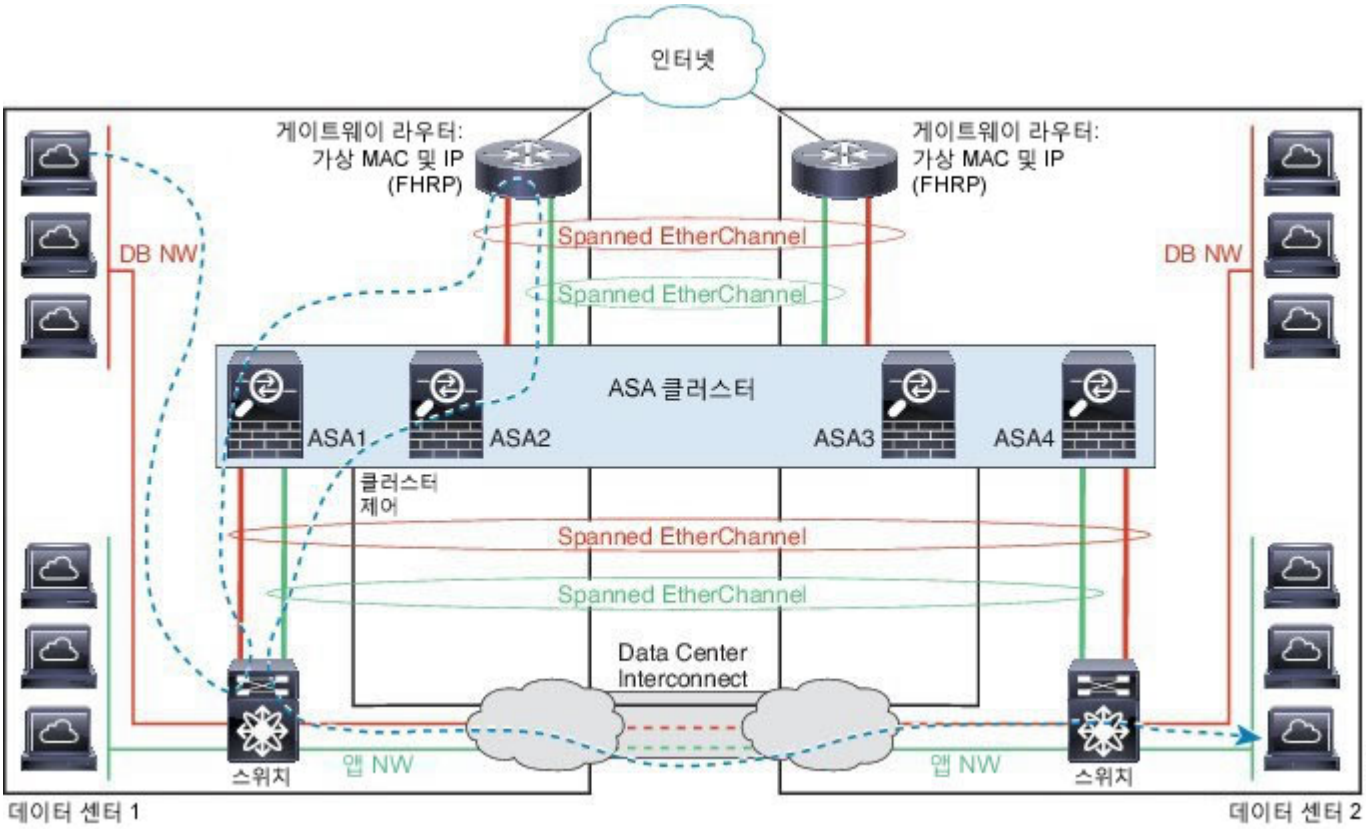


Spanned EtherChannel 투명 모드 East-West 사이트 간 예시

다음 예에서는 게이트웨이 라우터와 각 사이트의 두 내부 네트워크, 즉 애플리케이션 네트워크 및 DB 네트워크의 사이에 위치(East-West 삽입)한 2개 데이터 센터 각각에 2개의 클러스터 멤버가 있습니다.

클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부에 있는 애플리케이션 및 DB 네트워크에 대한 Spanned EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 섀시에 있습니다.

각 사이트의 게이트웨이 라우터는 HSRP와 같은 FHRP를 사용하여 각 사이트에 동일한 대상 가상 MAC 및 IP 주소를 제공합니다. 의도하지 않은 MAC 주소 플래핑을 방지하기 위한 좋은 방법은 **mac-address-table static** *outside_interface mac_address* 명령을 사용하여 게이트웨이 라우터의 실제 MAC 주소를 ASA MAC 주소 테이블에 정적으로 추가하는 것입니다. 이 항목이 없으면 사이트 1의 게이트웨이가 사이트 2의 게이트웨이와 통신할 경우, 해당 트래픽이 ASA를 통과하고 내부 인터페이스에서 사이트 2에 도달하려고 시도하여 문제가 발생할 수 있습니다. OTV(Overlay Transport Virtualization) 또는 이와 유사한 방법으로 데이터 VLAN이 사이트 전반에 확장됩니다. 트래픽이 게이트웨이 라우터로 향할 때 DCI를 통과하여 반대쪽 사이트에 가지 않도록 필터를 추가해야 합니다. 어떤 사이트의 게이트웨이 라우터가 연결할 수 없게 되면 트래픽이 다른 사이트의 게이트웨이 라우터에 전송될 수 있도록 필터를 제거해야 합니다.



vPC/VSS 옵션에 대한 자세한 내용은 Spanned EtherChannel 투명 모드 North-South 사이트 간 예시, 177 페이지의 내용을 참조하십시오.

클러스터링 기록

기능 이름	플랫폼릴리스	기능 정보
Cisco ASA를 위한 섀시 내 클러스터링	1.1.1	Firepower 9300 섀시 내에서 모든 ASA 보안 모듈을 클러스터링할 수 있습니다. 추가된 명령: **enter cluster-bootstrap, enter logical-device clustered, set chassis-id, set ipv4 gateway, set ipv4 pool, set ipv6 gateway, set ipv6 pool, set key, set mode spanned-etherchannel, set port-type cluster, set service-type, set virtual ipv4, set virtual ipv6**
6개의 ASA 모듈을 위한 섀시 간 클러스터링	1.1.3	이제 ASA를 위한 섀시 간 클러스터링을 활성화할 수 있습니다. 최대 6개의 섀시에 최대 6개의 모듈을 포함할 수 있습니다.
Firepower 9300의 Firepower Threat Defense에서 섀시 내 클러스터링 지원	1.1.4	Firepower 9300은 Firepower Threat Defense 애플리케이션이 있는 섀시 내 클러스터링을 지원합니다. 추가된 명령: **enter mgmt-bootstrap ftd, enter bootstrap-key FIREPOWER_MANAGER_IP, enter bootstrap-key FIREWALL_MODE, enter bootstrap-key-secret REGISTRATION_KEY, enter bootstrap-key-secret PASSWORD, enter bootstrap-key FQDN, enter bootstrap-key DNS_SERVERS, enter bootstrap-key SEARCH_DOMAINS, enter ipv4 firepower, enter ipv6 firepower, set value, set gateway, set ip, accept-license-agreement**
Firepower 4100/9300 섀시의 ASA에 대한 섀시 간 클러스터링 개선	2.1.1	이제 ASA 클러스터를 구축할 때 각 Firepower 4100/9300 섀시에 대해 사이트 ID를 구성할 수 있습니다. 이전에는 ASA 애플리케이션 내에서 사이트 ID를 구성해야 했습니다. 이 새로운 기능 덕분에 초기 구축이 쉬워졌습니다. 사이트 ID는 더 이상 ASA 컨피그레이션 내에서 설정할 수 없습니다. 또한 사이트 간 클러스터링과의 호환성을 최대한 활용하려면 안정성과 성능이 개선된 ASA 9.7(1) 및 FXOS 2.1.1로 업그레이드하는 것이 좋습니다. 수정된 명령: **set site-id**
6개의 Firepower Threat Defense 모듈을 위한 섀시 간 클러스터링	2.1.1	이제 Firepower Threat Defense를 위한 섀시 간 클러스터링을 활성화할 수 있습니다. 최대 6개의 섀시에 최대 6개의 모듈을 포함할 수 있습니다.

서비스 체이닝 구성

Cisco Firepower 4100/9300 섀시는 단일 블레이드에서 여러 서비스(예: 방화벽 및 서드파티 DDoS 애플리케이션)를 지원할 수 있습니다. 이러한 애플리케이션 및 서비스를 함께 연결하면 서비스 체인을 구성할 수 있습니다.

서비스 체이닝 정보

현재 지원되는 서비스 체이닝 컨피그레이션에서 서드파티 Radware DefensePro 가상 플랫폼은 ASA 방화벽보다 먼저 실행되거나 Firepower Threat Defense보다 먼저 실행되도록 설치될 수 있습니다. Radware DefensePro는 Firepower 4100/9300 섀시에서 DDoS(Distributed Denial-of-Service) 탐지 및 완화 기능을 제공하는 KVM 기반 가상 플랫폼입니다. 서비스 체이닝이 Firepower 4100/9300 섀시에서 활성화된 경우, 네트워크의 트래픽은 기본 ASA 또는 Firepower Threat Defense 방화벽에 도달하기 전에 먼저 DefensePro 가상 플랫폼을 통과해야 합니다.

Radware DefensePro 가상 플랫폼은 *Radware vDP*(가상 DefensePro) 또는 간단하게 *vDP*라고도 합니다. Radware DefensePro 가상 플랫폼은 경우에 따라 링크 데코레이터라고도 합니다.

서비스 체이닝 사전 요구 사항

Firepower 4100/9300 섀시에서 Radware DefensePro를 구축하기 전에 Firepower 4100/9300 섀시가 **etc/UTC** 표준 시간대와 함께 NTP 서버를 사용하도록 구성해야 합니다. Firepower 4100/9300 섀시의 날짜 및 시간 설정에 대한 자세한 내용은 날짜 및 시간 설정, 99 페이지의 내용을 참조하십시오.

서비스 체이닝 지침

모델

- Radware DefensePro 플랫폼은 Firepower 9300 보안 어플라이언스에서만 지원됩니다.
- Radware DefensePro 플랫폼은 다음 보안 어플라이언스의 Firepower Threat Defense에 지원됩니다.
  - Firepower 9300
  - Firepower 4110 - 데코레이터를 논리적 디바이스로 동시에 구축해야 합니다. 논리적 디바이스를 이 디바이스에서 이미 구성한 후에는 데코레이터를 설치할 수 없습니다.
  - Firepower 4120 - 데코레이터를 논리적 디바이스로 동시에 구축해야 합니다. 논리적 디바이스를 이 디바이스에서 이미 구성한 후에는 데코레이터를 설치할 수 없습니다.
  - Firepower 4140
  - Firepower 4150

추가 지침

- 서비스 체이닝은 섀시 간 클러스터 컨피그레이션에서 지원되지 않습니다. 그러나 Radware DefensePro 애플리케이션은 섀시 간 클러스터 시나리오의 독립형 컨피그레이션에서 구축할 수 있습니다.
- DefensePro 애플리케이션은 최대 3개의 보안 모듈에서 별도의 인스턴스로 실행할 수 있습니다.

독립형 논리적 디바이스에 **Radware DefensePro** 서비스 체인 구성

다음 절차에서는 Radware DefensePro를 독립형 ASA 또는 Firepower Threat Defense 논리적 디바이스보다 먼저 단일 서비스 체인에 설치하는 방법을 보여줍니다.

시작하기 전에

- Cisco.com에서 vDP 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 54 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 다운로드합니다(Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 57 페이지 참조).
- 섀시 내 클러스터의 독립형 컨피그레이션에서 Radware DefensePro 애플리케이션을 구축할 수 있습니다. 섀시 내 클러스터링에 대해서는 섀시 내 클러스터에 Radware DefensePro 서비스 체인 구성, 184 페이지의 내용을 참조하십시오.

절차

단계 **1** vDP에 대해 별도의 관리 인터페이스를 사용하려는 경우 인터페이스 속성 편집, 141 페이지에 따라 인터페이스를 활성화하고 관리 유형으로 설정합니다. 그렇지 않으면 애플리케이션 관리 인터페이스를 공유할 수 있습니다.

단계 **2** 독립형 컨피그레이션에서 ASA 또는 Firepower Threat Defense 논리적 디바이스를 생성(독립형 ASA 논리적 디바이스 생성, 148 페이지 또는독립형 Threat Defense 논리적 디바이스 생성, 150 페이지 참조)합니다. Firepower 4110 또는 4120 보안 어플라이언스에서 이미지를 설치 중인 경우, 컨피그레이션을 커밋하기 전에 Firepower Threat Defense 이미지와 함께 vDP를 설치해야 합니다.

단계 **3** 보안 서비스 모드를 시작합니다.
Firepower# **scopessa**

단계 **4** Radware vDP 인스턴스를 생성합니다.
Firepower /ssa # **scope slot** *slot_id*

Firepower /ssa/slot # **createapp-instance vdp**

Firepower /ssa/slot/app-instance* # **exit**

Firepower /ssa/slot/* # **exit**

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

단계 **6** 보안 모듈의 vDP 설치 및 프로비저닝을 확인합니다.
Firepower /ssa # **show app-instance**

예제:

```
Firepower /ssa # show app-instance
App Name    Slot ID    Admin State Oper State       Running Version Startup Version Cluster
 State    Cluster Role
---------- ---------- ----------- ---------------- --------------- ---------------
--------------- ------------
ftd         1          Enabled     Online           6.2.1.62        6.2.1.62        Not
Applicable  None
```

```
vdp         1          Disabled    Installing                      8.10.01.16-5    Not
Applicable  None
```

단계 **7** vDP 애플리케이션이 온라인 상태가 되고 나면 논리적 디바이스에 액세스합니다.
Firepower /ssa # **scopelogical-device** *device_name*

단계 **8** 관리 인터페이스를 vDP에 할당합니다. 논리적 디바이스의 경우와 동일한 물리적 인터페이스를 사용하거나 별도의 인터페이스를 사용할 수 있습니다.
Firepower /ssa/logical-device # **enterexternal-port-link** *nameinterface_id***vdp**

Firepower /ssa/logical-device/external-port-link* # **exit**

단계 **9** vDP용 외부 관리 인터페이스 설정을 구성합니다.

a) 부트스트랩 객체를 생성합니다.
Firepower /ssa/logical-device* # **create mgmt-bootstrap vdp**

b) 관리 IP 주소를 구성합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* #**createipv4** *slot_id***default**

c) 게이트웨이 주소를 설정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* #**setgateway** *gateway_address*

d) IP 주소 및 마스크를 설정합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* #**setip** *ip_address***mask** *network_mask*

e) 관리 IP 컨피그레이션 범위를 종료합니다.
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* #**exit**

f) 관리 부트스트랩 컨피그레이션 범위를 종료합니다.
Firepower /ssa/logical-device/mgmt-bootstrap* #**exit**

단계 **10** ASA 또는 Firepower Threat Defense 플로우보다 먼저 vDP를 배치할 위치의 데이터 인터페이스를 편집합니다.
Firepower /ssa/logical-device* # **scopeexternal-port-link** *name*

show external-port-link 명령을 입력하여 인터페이스 이름을 봅니다.

단계 **11** 논리적 디바이스에 vDP를 추가합니다.
Firepower /ssa/logical-device/external-port-link* # **setdecorator vdp**

vDP를 사용할 각 인터페이스에 대해 반복합니다.

단계 **12** 서드파티 앱이 인터페이스에 대해 설정되었는지 확인합니다.
Firepower /ssa/logical-device/external-port-link* # **show detail**

예제:

```
Firepower /ssa/logical-device/external-port-link # show detail

External-Port Link:
    Name: Ethernet11_ftd
    Port or Port Channel Name: Ethernet1/1
    App Name: ftd
    Description:
    Link Decorator: vdp
```

단계 **13** 컨피그레이션을 커밋합니다.
commit-buffer

섀시 내 클러스터에 **Radware DefensePro** 서비스 체인 구성

시작하기 전에

- Cisco.com에서 vDP 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 54 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 다운로드합니다(Firepower 4100/9300 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 57 페이지 참조).

절차

단계 **1** vDP에 대해 별도의 관리 인터페이스를 사용하려는 경우 인터페이스 속성 편집, 141 페이지에 따라 인터페이스를 활성화하고 관리 유형으로 설정합니다. 그렇지 않으면 애플리케이션 관리 인터페이스를 공유할 수 있습니다.

단계 **2** ASA 섀시 내 클러스터(ASA 클러스터링 구성, 163 페이지 참조) 또는 Firepower Threat Defense 섀시 내 클러스터(Firepower Threat Defense 클러스터링 구성, 169 페이지 참조)를 구성합니다.

단계 **3** Radware DefensePro를 사용하여 외부(클라이언트 대상) 포트를 데코레이팅합니다.
enter external-port-link *name interface_name* { **asa** | **ftd**}

set **decoratorvdp**

set **description""**

exit

단계 **4** 논리적 디바이스용 외부 관리 포트를 할당합니다.
enter **external-port-link** *{ mgmt_asa* | *mgmt_ftd } interface_id { asa* | *ftd }*

set **decorator""**

set **description""**

exit

단계 **5** DefensePro용 외부 관리 포트를 할당합니다.
enter **external-port-linkmgmt_vdp** *interface_name { asa* | *ftd }*

set **decorator""**

set **description""**

단계 **6** 클러스터 포트 채널을 구성합니다.
enter **external-port-link** port-channel48 Port-channel48 *{ asa* | *ftd }*

set **decorator""**

set **description""**

exit

단계 **7** 3개의 DefensePro 인스턴스 모두에 대해 관리 부트스트랩을 구성합니다.
enter **mgmt-bootstrapvdp**

enter **ipv4** *slot_id***default**

setgateway *gateway_address*

setip *ip_address***mask** *network_mask*

exit

예제:

```
enter mgmt-bootstrap vdp
    enter ipv4 1 default
        set gateway 172.16.0.1
        set ip 172.16.4.219 mask 255.255.0.0
    exit

    enter ipv4 2 default
        set gateway 172.16.0.1
        set ip 172.16.4.220 mask 255.255.0.0
    exit

    enter ipv4 3 default
        set gateway 172.16.0.1
        set ip 172.16.4.221 mask 255.255.0.0
    exit
```

단계 **8** 관리 부트스트랩 컨피그레이션 범위를 종료합니다.
exit

단계 **9** 마스터 블레이드에서 관리 IP를 설정하고 클러스터링을 활성화합니다.
deviceclusteringmanagement-channelip

deviceclusteringmasterset *management-channel ip*

deviceclusteringstatesetenable

단계 **10** 컨피그레이션을 커밋합니다.
commit-buffer

참고 이 절차를 완료한 후 DefensePro 인스턴스가 클러스터에 구성되었는지 확인해야 합니다.

단계 **11** 다음 방법 중 하나를 사용하여 어떤 DefensePro 인스턴스가 기본 또는 보조인지 확인합니다.

a) DefensePro 인스턴스 범위를 설정하고 DefensePro의 애플리케이션 속성만 표시합니다.
scopessa

scope *slot_number*

scopeapp-instancevdp

showapp-attri

b) 슬롯 범위를 설정하고 DefensePro 인스턴스 정보를 더 자세히 표시합니다. 이 접근 방식을 사용하면 슬롯에 있는 논리적 디바이스 및 vDP 애플리케이션 인스턴스의 정보가 모두 표시됩니다.
scopessa

scope *slot_number*

showapp-instance expand detail

DefensePro 애플리케이션이 온라인 상태지만 클러스터에는 아직 구성되지 않은 경우 CLI에서 다음을 표시합니다.

```
    App Attribute:
        App Attribute Key: cluster-role
        Value: unknown
```

시스템이 "알 수 없음" 값을 표시하면 DefensePro 애플리케이션을 시작하고 vDP 클러스터를 생성하도록 마스터 IP 주소를 구성해야 합니다.

DefensePro 애플리케이션이 온라인 상태이며 클러스터에 구성되어 있는 경우 CLI에서는 다음을 표시합니다.

```
    App Attribute:
        App Attribute Key: cluster-role
        Value: primary/secondary
```

예

```
scope ssa
  enter logical-device ld asa "1,2,3" clustered
     enter cluster-bootstrap
         set chassis-id 1
         set ipv4 gateway 172.16.0.1
         set ipv4 pool 172.16.4.216 172.16.4.218
         set ipv6 gateway 2010::2
         set ipv6 pool 2010::21 2010::26
         set key secret
         set mode spanned-etherchannel
         set name cisco
         set virtual ipv4 172.16.4.222 mask 255.255.0.0
         set virtual ipv6 2010::134 prefix-length 64
     exit
     enter external-port-link Ethernet1-2 Ethernet1/2 asa
         set decorator vdp
         set description ""
     exit
     enter external-port-link Ethernet1-3_asa Ethernet1/3 asa
         set decorator ""
         set description ""
     exit
     enter external-port-link mgmt_asa Ethernet1/1 asa
         set decorator ""
         set description ""
     exit
     enter external-port-link mgmt_vdp Ethernet1/1 vdp
         set decorator ""
         set description ""
     exit
     enter external-port-link port-channel48 Port-channel48 asa
         set decorator ""
         set description ""
     exit
     enter mgmt-bootstrap vdp
         enter ipv4 1 default
             set gateway 172.16.0.1
             set ip 172.16.4.219 mask 255.255.0.0
         exit

         enter ipv4 2 default
             set gateway 172.16.0.1
             set ip 172.16.4.220 mask 255.255.0.0
         exit

         enter ipv4 3 default
             set gateway 172.16.0.1
             set ip 172.16.4.221 mask 255.255.0.0
         exit
 exit
commit-buffer
scope ssa
```

```
scope slot 1
scope app-instance vdp
show app-attri
 App Attribute:
     App Attribute Key: cluster-role
     Value: unknown
```

UDP/TCP 포트 열기 및 vDP 웹 서비스 활성화

Radware APSolute Vision Manager 인터페이스는 다양한 UDP/TCP 포트를 사용하여 Radware vDP 애플리케이션과 통신합니다. vDP 애플리케이션이 APSolute Vision Manager와 통신하기 위해서는 이러한 포트에 액세스할 수 있으며 포트가 방화벽에 의해 차단되지 않음을 확인해야 합니다. 열어야 할 특정 포트에 대한 자세한 내용은 APSolute Vision 사용자 가이드의 다음 표를 참조하십시오.

- **APSolute Vision Server-WBM** 통신용 포트 및 운영 체제
- **Radware** 디바이스를 사용하는 **APSolute Vision Server**용 통신 포트

Radware APSolute Vision에서 FXOS 섀시에 구축된 가상 DefensePro 애플리케이션을 관리하려면 FXOS CLI를 사용하여 vDP 웹 서비스를 활성화해야 합니다.

절차

단계 **1** FXOS CLI에서 vDP 애플리케이션 인스턴스에 연결합니다.
connect module *slot***console**

connect vdp

단계 **2** vDP 웹 서비스를 활성화합니다.
manage secure-web status set enable

단계 **3** vDP 애플리케이션 콘솔을 종료하고 FXOS 모듈 CLI로 돌아갑니다.
Ctrl]

논리적 디바이스 관리

논리적 디바이스를 삭제하고, ASA를 투명 모드로 변환하고, 인터페이스 컨피그레이션을 변경할 수 있으며 기존의 논리적 디바이스에서 다른 작업을 수행할 수 있습니다.

애플리케이션 또는 데코레이터 콘솔에 연결

다음 절차를 사용하여 애플리케이션 또는 데코레이터 콘솔에 연결합니다.

참고 콘솔 액세스 시 문제가 발생한 경우, 다른 SSH 클라이언트를 시도하거나 SSH 클라이언트를 새 버전으로 업그레이드하는 것이 좋습니다.

절차

단계 **1** 애플리케이션 또는 데코레이터 콘솔에 연결하려면 다음을 수행합니다.

a) FXOS CLI에서 보안 모듈/엔진에 연결합니다.
Firepower-chassis # **connectmodule** *slot_number***console**
참고 여러 보안 모듈을 지원하지 않는 디바이스의 보안 엔진에 연결하려면 *slot_number*에 1을 사용합니다.
보안 모듈에 처음 연결하는 경우 FXOS 모듈 CLI에 액세스합니다.

b) 애플리케이션 또는 데코레이터에 연결하려면 디바이스에 적절한 명령을 입력합니다. Firepower-module1>**connect asa**
Firepower-module1>**connect ftd**
Firepower-module1>**connect vdp**
FXOS CLI의 수퍼바이저 레벨에서 보안 모듈/엔진에 대한 후속 연결은 보안 모듈/엔진 OS에 직접액세스됩니다.=

단계 **2** (선택 사항) **Ctrl-A-D**를 입력하여 FXOS 모듈 CLI에 대한 애플리케이션 콘솔을 종료합니다. **Ctrl-]-**를 입력하여 FXOS 모듈 CLI에 대한 데코레이터 콘솔을 종료합니다.
트러블슈팅을 위해 FXOS 모듈 CLI에 액세스할 수 있습니다.

단계 **3** FXOS CLI의 수퍼바이저 레벨로 돌아갑니다.

a) 보안 모듈/엔진 콘솔을 종료하려면 ~를 입력합니다.
텔넷 애플리케이션을 종료합니다.

b) 텔넷 애플리케이션을 종료하려면 다음을 입력합니다.
telnet>**quit**

예

다음 예에서는 보안 모듈 1에 있는 ASA에 연결한 다음 종료하여 FXOS CLI의 수퍼바이저 레벨로 다시 돌아갑니다.

```
Firepower# connect module 1 console
Telnet escape character is '~'.
Trying 127.5.1.1...
Connected to 127.5.1.1.
Escape character is '~'.

CISCO Serial Over LAN:
Close Network Connection to Exit

Firepower-module1>connect asa
```

```
asa> ~
telnet> quit
Connection closed.
Firepower#
```

논리적 디바이스 삭제

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower# **scopessa**

단계 **2** 섀시에 있는 논리적 디바이스에 대한 세부사항을 확인합니다.
Firepower /ssa # **showlogical-device**

단계 **3** 삭제할 각 논리적 디바이스에 대해 다음 명령을 입력합니다.
Firepower /ssa # **deletelogical-device** *device_name*

단계 **4** 논리적 디바이스에 설치된 애플리케이션에 대한 세부사항을 확인합니다.
Firepower /ssa # **showapp-instance**

단계 **5** 삭제할 애플리케이션 각각에 대해 다음 명령을 입력합니다.

a) Firepower /ssa # **scopeslot** *slot_number*
b) Firepower /ssa/slot # **deleteapp-instance** *application_name*
c) Firepower /ssa/slot # **exit**

단계 **6** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa # show logical-device

Logical Device:
    Name       Description Slot ID    Mode       Operational State        Template Name
    ---------- ----------- ---------- ---------- ------------------------ -------------
    FTD                    1,2,3      Clustered  Ok                       ftd
Firepower /ssa # delete logical-device FTD
Firepower /ssa* # show app-instance
Application Name     Slot ID     Admin State     Operational State    Running Version Startup
 Version Cluster Oper State
-------------------- ----------- --------------- -------------------- ---------------
--------------- ------------------
ftd                              1 Disabled        Stopping             6.0.0.837
6.0.0.837       Not Applicable
ftd                              2 Disabled        Offline              6.0.0.837
6.0.0.837       Not Applicable
ftd                              3 Disabled        Not Available
6.0.0.837       Not Applicable
Firepower /ssa* # scope slot 1
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
```

```
Firepower /ssa* # scope slot 2
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 3
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # commit-buffer
```

논리적 디바이스와 연결되지 않은 애플리케이션 인스턴스 삭제

논리적 디바이스를 삭제하는 경우, 논리적 디바이스에 대한 애플리케이션 컨피그레이션도 삭제할지를 묻는 프롬프트가 표시됩니다. 애플리케이션 컨피그레이션을 삭제하지 않는 경우, 해당 애플리케이션 인스턴스를 삭제할 때까지 다른 애플리케이션을 사용하여 논리적 디바이스를 생성할 수 없습니다. 애플리케이션 인스턴스가 더 이상 논리적 디바이스와 연결된 상태가 아닌 경우 다음 절차를 사용하여 보안 모듈/엔진에서 애플리케이션 인스턴스를 삭제할 수 있습니다.

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower# **scopessa**

단계 **2** 설치된 애플리케이션에 대한 세부사항을 확인합니다.
Firepower /ssa # **showapp-instance**

단계 **3** 삭제할 애플리케이션 각각에 대해 다음 명령을 입력합니다.

a) Firepower /ssa # **scopeslot** *slot_number*
b) Firepower /ssa/slot # **deleteapp-instance** *application_name*
c) Firepower /ssa/slot # **exit**

단계 **4** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa* # show app-instance
Application Name    Slot ID    Admin State    Operational State    Running Version Startup
 Version Cluster Oper State
------------------- ---------- -------------- -------------------- ---------------
--------------- ------------------
ftd                           1 Disabled       Stopping             6.0.0.837
6.0.0.837       Not Applicable
ftd                           2 Disabled       Offline              6.0.0.837
6.0.0.837       Not Applicable
ftd                           3 Disabled       Not Available
6.0.0.837       Not Applicable
Firepower /ssa* # scope slot 1
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 2
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 3
```

```
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # commit-buffer
```

Firepower Threat Defense 논리적 디바이스에서 인터페이스 변경

Firepower Threat Defense 논리적 디바이스에서 인터페이스를 할당 또는 할당 해제할 수 있습니다. 그러면 Firepower Management Center에서 인터페이스 컨피그레이션을 동기화할 수 있습니다.

시작하기 전에

- 인터페이스를 구성하고 인터페이스 속성 편집, 141 페이지 및 포트 채널 생성, 142 페이지에 따라 임의의 EtherChannels을 추가합니다.
- 논리적 디바이스에 영향을 미치거나 Firepower Management Center에서 동기화를 요구하지 않고 할당된 EtherChannel의 멤버십을 편집할 수 있습니다.
- 이미 할당된 인터페이스를 EtherChannel에 추가하려는 경우(예: 모든 인터페이스가 기본적으로 클러스터에 할당됨), 먼저 논리적 디바이스에서 인터페이스의 할당을 해제한 다음 해당 인터페이스를 EtherChannel에 추가해야 합니다. 새 EtherChannel의 경우, 그런 다음 EtherChannel을 디바이스에 할당할 수 있습니다.
- 관리 또는 Firepower 이벤트 인터페이스를 교체하려는 경우, Firepower Chassis Manager를 사용해야 합니다. CLI는 이 변경 사항을 지원하지 않습니다.
- 클러스터링 또는 고가용성을 위해, Firepower Management Center에서 컨피그레이션을 동기화하기 전에 모든 유닛에서 인터페이스를 추가하거나 제거해야 합니다. 새 인터페이스는 관리가 다운된 상태에서 추가되므로 인터페이스 모니터링에는 영향을 미치지 않습니다.

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower# **scopessa**

단계 **2** 논리적 디바이스를 편집합니다.
Firepower /ssa # **scopelogical-device** *device_name*

단계 **3** 논리적 디바이스에서 인터페이스의 할당을 해제합니다.
Firepower /ssa/logical-device # **deleteexternal-port-link** *name*

show external-port-link 명령을 입력하여 인터페이스 이름을 봅니다.

단계 **4** 논리적 디바이스에 새 인터페이스를 할당합니다.
Firepower /ssa/logical-device* # **createexternal-port-link** *name interface_id***ftd**

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

단계 **6** Firepower Management Center에 로그인합니다.

단계 **7** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택하고 Firepower Threat Defense 디바이스의 편집 아이콘( )을 클릭합니다. 기본적으로는 **Interfaces(**인터페이스**)** 탭이 선택됩니다.

단계 **8** **Interfaces(**인터페이스**)** 탭의 왼쪽 상단에 있는 **Sync Interfaces from device(**디바이스에서 인터페이스 동기화**)** 버튼을 클릭합니다.

단계 **9** **Save(**저장**)**를 클릭합니다.
이제 **Deploy(**구축**)**를 클릭하고 할당된 디바이스에 정책을 구축할 수 있습니다. 변경 사항은 구축할 때까지 활성화되지 않습니다.

ASA 논리적 디바이스에서 인터페이스 변경

ASA 논리적 디바이스에서 관리 인터페이스를 할당, 할당 해제 또는 교체할 수 있습니다. ASDM은 새로운 인터페이스를 자동으로 검색합니다.

시작하기 전에

- 인터페이스를 구성하고 인터페이스 속성 편집, 141 페이지 및 포트 채널 생성, 142 페이지에 따라 임의의 EtherChannels을 추가합니다.
- 논리적 디바이스에 영향을 미치지 않고 할당된 EtherChannel의 멤버십을 편집할 수 있습니다.
- 이미 할당된 인터페이스를 EtherChannel에 추가하려는 경우(예: 모든 인터페이스가 기본적으로 클러스터에 할당됨), 먼저 논리적 디바이스에서 인터페이스의 할당을 해제한 다음 해당 인터페이스를 EtherChannel에 추가해야 합니다. 새 EtherChannel의 경우, 그런 다음 EtherChannel을 디바이스에 할당할 수 있습니다.
- 클러스터링 또는 장애 조치의 경우 새 인터페이스는 관리가 다운된 상태에서 추가되므로 인터페이스 모니터링에는 영향을 미치지 않습니다.

절차

단계 **1** 보안 서비스 모드를 시작합니다.
Firepower# **scopessa**

단계 **2** 논리적 디바이스를 편집합니다.
Firepower /ssa # **scopelogical-device** *device_name*

단계 **3** 논리적 디바이스에서 인터페이스의 할당을 해제합니다.
Firepower /ssa/logical-device # **deleteexternal-port-link** *name*

show external-port-link 명령을 입력하여 인터페이스 이름을 봅니다.

관리 인터페이스의 경우 새로운 관리 인터페이스를 추가하기 전에 현재 인터페이스를 삭제한 다음 **commit-buffer** 명령을 사용하여 변경 사항을 커밋합니다.

단계 **4** 논리적 디바이스에 새 인터페이스를 할당합니다.
Firepower /ssa/logical-device* # **createexternal-port-link** *name interface_id***asa**

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

12 장

컨피그레이션 가져오기/내보내기

컨피그레이션 가져오기/내보내기 정보

컨피그레이션 내보내기 기능을 사용하여 Firepower 4100/9300 섀시의 논리적 디바이스와 플랫폼 컨피그레이션 설정이 포함된 XML 파일을 원격 서버에 내보낼 수 있습니다. 나중에 해당 컨피그레이션 파일을 가져와서 컨피그레이션 설정을 Firepower 4100/9300 섀시에 신속하게 적용하여 알려진 정상적인 컨피그레이션으로 돌아가거나 시스템 장애를 복구할 수 있습니다.

지침 및 제한 사항

- 컨피그레이션 파일의 콘텐츠를 수정하지 마십시오. 컨피그레이션 파일이 수정된 경우, 해당 파일을 사용하는 컨피그레이션 가져오기가 실패할 수 있습니다.
- 애플리케이션별 컨피그레이션 설정은 컨피그레이션 파일에 포함되어 있지 않습니다. 애플리케이션별 설정 및 컨피그레이션을 관리하기 위해 애플리케이션에서 제공하는 컨피그레이션 백업 툴을 사용해야 합니다.
- 컨피그레이션을 Firepower 4100/9300 섀시에 가져올 때 Firepower 4100/9300 섀시의 모든 기존 컨피그레이션(모든 논리적 디바이스 포함)이 삭제되며 가져오기 파일에 포함된 컨피그레이션으로 완전히 대체됩니다.
- 컨피그레이션을 내보낸 동일한 Firepower 4100/9300 섀시에만 컨피그레이션 파일을 가져오는 것이 좋습니다.
- 가져오기 작업 중인 Firepower 4100/9300 섀시의 플랫폼 소프트웨어 버전은 내보내기를 수행할 때와 동일한 버전이어야 합니다. 그렇지 않으면 가져오기 작업의 성공이 보장되지 않습니다.

Firepower 4100/9300 섀시가 업그레이드 또는 다운그레이드될 때마다 백업 컨피그레이션을 내보내는 것이 좋습니다.

- 가져오기 작업 중인 Firepower 4100/9300 섀시에는 내보내기를 수행할 때와 동일한 슬롯에 동일한 네트워크 모듈이 설치되어 있어야 합니다.
- 가져오기 작업 중인 Firepower 4100/9300 섀시에는 가져오기 작업 중인 내보내기 파일에 정의된 모든 논리적 디바이스에 올바른 소프트웨어 애플리케이션 이미지가 설치되어 있어야 합니다.
- 애플리케이션에 EULA(엔드 유저 라이선스 계약)가 있는 논리적 디바이스가 가져오기 작업 중인 컨피그레이션 파일에 포함된 경우, 컨피그레이션 가져오기 또는 작업이 실패하기 전에 해당 애플리케이션에 대한 EULA가 Firepower 4100/9300 섀시에서 허용되어야 합니다.
- 기존 백업 파일의 덮어쓰기를 방지하려면 백업 작업 시 파일 이름을 변경하거나 기존 파일을 다른 위치에 복사하십시오.

컨피그레이션 파일 내보내기

컨피그레이션 내보내기 기능을 사용하여 Firepower 4100/9300 섀시의 논리적 디바이스와 플랫폼 컨피그레이션 설정이 포함된 XML 파일을 원격 서버에 내보냅니다.

컨피그레이션 내보내기 기능 사용에 대한 중요한 정보는 컨피그레이션 가져오기/내보내기 정보을 참조하십시오.

절차

단계 **1** 컨피그레이션 파일을 원격 서버에 내보내려면 다음을 수행합니다.
scopesystem
export-config *URL* **enabled commit-buffer**

다음 구문 중 하나를 사용하여 내보낼 파일의 URL을 지정합니다.

- **ftp**://*username@hostname/path/image_name*
- **scp**://*username@hostname/path/image_name*
- **sftp**://*username@hostname/path/image_name*
- **tftp**://*hostname:port-num/path/image_name*

참고 파일 이름이 포함된 전체 경로를 지정해야 합니다. 파일 이름을 지정하지 않는 경우, 지정된 경로에 숨겨진 파일이 생성됩니다.

예제:
```
Firepower-chassis# scope system
Firepower-chassis /system # export-config scp://user1@192.168.1.2:/export/cfg-backup.xml
enabled
Firepower-chassis /system/export-config # commit-buffer
```

단계 **2** 다음 명령을 사용하여 내보내기 작업의 상태를 확인합니다.
scopesystem

scope export-config *hostname*

show fsm status

예제:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope export-config 192.168.1.2
Firepower-chassis /system/export-config # show fsm status

Hostname: 192.168.1.2

    FSM 1:
        Remote Result: Not Applicable
        Remote Error Code: None
        Remote Error Description:
        Status: Nop
        Previous Status: Backup Success
        Timestamp: 2016-01-03T15:32:08.636
        Try: 0
        Progress (%): 100
        Current Task:
```

단계 **3** 다음 명령을 사용하여 기존의 내보내기 작업을 확인합니다.
scopesystem
show export-config

단계 **4** 다음 명령을 사용하여 기존의 내보내기 작업을 수정합니다.
scopesystem
scope export-config *hostname*

다음 명령을 사용하여 내보내기 작업을 수정합니다.

- {**enable**|**disable**}
- **set description** <*description*>
- **set password** <*password*>
- **set port** <*port*>
- **set protocol** {**ftp**|**scp**|**sftp**|**tftp**}
- **set remote-file** *path_and_filename*
- **set user** <*user*>

단계 **5** 다음 명령을 사용하여 내보내기 작업을 삭제합니다.
scopesystem
delete export-config *hostname*

commit-buffer

자동 컨피그레이션 내보내기 예약

예약된 내보내기 기능을 사용하여 Firepower 4100/9300 섀시의 논리적 디바이스와 플랫폼 컨피그레이션 설정이 포함된 XML 파일을 원격 서버에 자동으로 내보냅니다. 매일, 매주 또는 2주마다 실행되도록 내보내기를 예약할 수 있습니다. 예약된 내보내기 기능이 활성화된 시기를 기준으로 예약에 따라 컨피그레이션 내보내기가 실행됩니다. 예를 들어 매주 수요일 오후 10시에 내보내기를 예약한 경우 시스템은 수요일마다 오후 10시에 새로운 내보내기를 트리거합니다.

컨피그레이션 내보내기 기능 사용에 대한 중요한 정보는 컨피그레이션 가져오기/내보내기 정보을 참조하십시오.

절차

예약된 내보내기 작업을 생성하려면 다음을 수행합니다.

a) 정책 컨피그레이션을 내보내도록 범위를 설정합니다.
scopeorg
scope cfg-export-policy default

b) 내보내기 정책을 활성화합니다.
setadminstate enable

c) 원격 서버와 통신할 때 사용할 프로토콜을 지정합니다.
set protocol {**ftp**|**scp**|**sftp**|**tftp**}

d) 백업 파일을 저장해야 하는 위치의 IP 주소 또는 호스트 이름을 지정합니다. 이는 Firepower 4100/9300 섀시가 네트워크를 통해 액세스할 수 있는 서버, 스토리지 어레이, 로컬 드라이브 또는 읽기/쓰기 미디어일 수 있습니다.
IP 주소 대신 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다.

sethostname *hostname*

e) 기본 포트 이외의 포트를 사용 중인 경우, 포트 번호를 지정합니다.
setport *port*

f) 원격 서버에 로그인할 때 시스템에서 사용해야 하는 사용자 이름을 지정합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.
setuser *username*

g) 원격 서버 사용자 이름의 비밀번호를 지정합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.
setpassword *password*

h) 파일 이름을 포함하여 컨피그레이션 파일을 내보낼 전체 경로를 지정합니다. 파일 이름을 생략하는 경우 내보내기 절차에서 파일에 이름이 할당됩니다.
setremote-file *path_and_filename*

i) 컨피그레이션을 자동으로 내보낼 일정을 지정합니다. 이는 Daily(매일), Weekly(매주) 또는 BiWeekly(격주) 중 하나일 수 있습니다.
setschedule {**daily**|**weekly**|**bi-weekly**}

j) 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

commit-buffer

예제:

```
Firepower-chassis# scope org
Firepower-chassis /org # scope cfg-export-policy default
Firepower-chassis /org/cfg-export-policy # set adminstate enable
Firepower-chassis /org/cfg-export-policy* # set protocol scp
Firepower-chassis /org/cfg-export-policy* # set hostname 192.168.1.2
Firepower-chassis /org/cfg-export-policy* # set remote-file /export/cfg-backup.xml
Firepower-chassis /org/cfg-export-policy* # set user user1
Firepower-chassis /org/cfg-export-policy* # set password
Password:
Firepower-chassis /org/cfg-export-policy* # set schedule weekly
Firepower-chassis /org/cfg-export-policy* # commit-buffer
Firepower-chassis /org/cfg-export-policy #
Firepower-chassis /org/cfg-export-policy # show detail

Config Export policy:
    Name: default
    Description: Configuration Export Policy
    Admin State: Enable
    Protocol: Scp
    Hostname: 192.168.1.2
    User: user1
    Remote File: /export/cfg-backup.xml
    Schedule: Weekly
    Port: Default
    Current Task:
```

컨피그레이션 내보내기 미리 알림 설정

특정 기간(일수)에 컨피그레이션 내보내기가 실행되지 않은 경우 시스템에서 결함을 생성하도록 하려면 Export Reminder(내보내기 미리 알림) 기능을 사용합니다.

절차

다음 명령을 사용하여 컨피그레이션 내보내기 미리 알림을 생성합니다.

scopeorg

scope cfg-export-reminder

set frequency *days*

set adminstate {**enable**|**disable**}

commit-buffer

예제:

```
Firepower-chassis# scope org
Firepower-chassis /org # scope cfg-export-reminder
Firepower-chassis /org/cfg-export-reminder # set frequency 10
Firepower-chassis /org/cfg-export-reminder* # set adminstate enable
Firepower-chassis /org/cfg-export-reminder* # commit-buffer
Firepower-chassis /org/cfg-export-reminder # show detail

Config Export Reminder:
    Config Export Reminder (Days): 10
    AdminState: Enable
```

컨피그레이션 파일 가져오기

컨피그레이션 가져오기 기능을 사용하여 Firepower 4100/9300 섀시에서 이전에 내보낸 컨피그레이션 설정을 적용할 수 있습니다. 이 기능을 사용하면 알려진 정상적인 컨피그레이션으로 돌아가거나 시스템 장애를 복구할 수 있습니다. 컨피그레이션 가져오기 기능 사용에 대한 중요한 정보를 보려면 컨피그레이션 가져오기/내보내기 정보을 검토하십시오.

절차

단계 **1** 다음 명령을 사용하여 원격 서버에서 컨피그레이션 파일을 가져옵니다.
scopesystem
import-config *URL* **enabled**

commit-buffer

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*/*image_name*
- **scp**://*username@hostname*/*path*/*image_name*
- **sftp**://*username@hostname*/*path*/*image_name*
- **tftp**://*hostname*:*port-num*/*path*/*image_name*

예제:

```
Firepower-chassis# scope system
Firepower-chassis /system # import-config scp://user1@192.168.1.2:/import/cfg-backup.xml
enabled
Warning: After configuration import any changes on the breakout port configuration will
  cause the system to reboot
Firepower-chassis /system/import-config # commit-buffer
```

단계 **2** 다음 명령을 사용하여 가져오기 작업의 상태를 확인합니다.
scopesystem
scope import-config *hostname*

show fsm status

예제:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope import-config 192.168.1.2
Firepower-chassis /system/import-config # show fsm status

Hostname: 192.168.1.2

    FSM 1:
        Remote Result: Not Applicable
        Remote Error Code: None
        Remote Error Description:
        Status: Import Wait For Switch
        Previous Status: Import Config Breakout
        Timestamp: 2016-01-03T15:45:03.963
        Try: 0
        Progress (%): 97
```

```
Current Task: updating breakout port configuration(FSM-STAGE:sam:dme:
  MgmtImporterImport:configBreakout)
```

단계 **3** 다음 명령을 사용하여 기존 가져오기 작업을 확인합니다.
scopesystem
show import-config

단계 **4** 다음 명령을 사용하여 기존 가져오기 작업을 수정합니다.
scopesystem
scope import-config *hostname*

다음 명령을 사용하여 가져오기 작업을 수정합니다.

- {**enable**|**disable**}
- **set description** <*description*>
- **set password** <*password*>
- **set port** <*port*>
- **set protocol** {**ftp**|**scp**|**sftp**|**tftp**}
- **set remote-file** *path_and_filename*
- **set user** <*user*>

단계 **5** 다음 명령을 사용하여 가져오기 작업을 삭제합니다.
scopesystem
delete import-config *hostname*

commit-buffer

13 장

트러블슈팅

패킷 캡처

패킷 캡처 툴은 연결 및 컨피그레이션 문제를 디버깅하는 데 사용되는 중요한 자산이며 Firepower 4100/9300 섀시를 통과하는 트래픽 플로우를 파악하기 위해 사용됩니다. 패킷 캡처 툴을 사용하여 Firepower 4100/9300 섀시에서 특정 고객 대상 포트 또는 애플리케이션 포트를 통과하는 트래픽을 기록할 수 있습니다.

여러 개의 패킷 캡처 세션을 생성할 수 있으며 각 세션에서는 여러 포트의 트래픽을 캡처할 수 있습니다. 패킷 캡처 세션에 포함된 각 포트에 대해 별도의 패킷 캡처(PCAP) 파일이 생성됩니다.

백플레인 포트 매핑

Firepower 4100/9300 섀시는 내부 백플레인 포트에 다음 매핑을 사용합니다.

보안 모듈	포트 매핑	설명
보안 모듈 1/보안 엔진	Ethernet1/9	Internal-Data0/0
보안 모듈 1/보안 엔진	Ethernet1/10	Internal-Data0/1
보안 모듈 2	Ethernet1/11	Internal-Data0/0
보안 모듈 2	Ethernet1/12	Internal-Data0/1

보안 모듈	포트 매핑	설명
보안 모듈 3	Ethernet1/13	Internal-Data0/0
보안 모듈 3	Ethernet1/14	Internal-Data0/1

지침 및 제한 사항

패킷 캡처 툴에는 다음과 같은 제한 사항이 있습니다.

- 최대 100Mbps만 캡처할 수 있습니다.
- 패킷 캡처 세션을 실행하는 데 사용할 수 있는 스토리지 공간이 충분하지 않은 경우에도 패킷 캡처 세션을 생성할 수 있습니다. 패킷 캡처 세션을 시작하기 전에 사용할 수 있는 충분한 스토리지 공간이 있는지 확인해야 합니다.
- 여러 활성 패킷 캡처 세션은 지원되지 않습니다.
- 내부 스위치의 인그레스 단계에서만 캡처합니다.
- 내부 스위치에서 파악할 수 없는 패킷에서는 필터가 유효하지 않습니다(예: 보안 그룹 태그 및 네트워크 서비스 헤더 패킷).
- 추상화가 지원되지 않습니다(예: 포트 채널 및 서비스 체인).

참고 포트 채널에서 트래픽 캡처가 지원되지는 않지만 패킷 캡처 세션에서 포트 채널을 구성하는 개별 멤버 포트를 포함할 수는 있으며, 패킷 캡처 툴은 각 해당 멤버 포트에 대해 별도의 패킷 캡처 파일을 생성합니다.

- 캡처 세션이 계속 활성 상태인 경우 PCAP 파일을 복사하거나 내보낼 수 없습니다.
- 패킷 캡처 세션을 삭제하면 해당 세션에 연결된 패킷 캡처 파일도 모두 삭제됩니다.

패킷 캡처 세션 생성 또는 편집

절차

단계 **1** 캡처 모드를 시작합니다.
Firepower-chassis # **scope packet-capture**

단계 **2** 새 패킷 캡처 세션을 생성합니다.
Firepower-chassis /packet-capture # **create session** *session_name*

기존의 패킷 캡처 세션을 편집합니다.

Firepower-chassis /packet-capture # **enter session** *session_name*

단계 **3** 이 패킷 캡처 세션에 사용할 버퍼 크기를 지정합니다.
Firepower-chassis /packet-capture/session # **set session-memory-usage** *session_size_in_megabytes*

지정된 버퍼 크기는 1~2,048MB여야 합니다.

단계 **4** 이 패킷 캡처 세션용으로 캡처할 패킷의 길이를 지정합니다.
Firepower-chassis /packet-capture/session # **set session-pcap-snaplength** *session_snap_length_in_bytes*

지정된 스냅 길이는 64~9006바이트여야 합니다. 세션 스냅 길이를 구성하지 않는 경우 기본 캡처 길이는 1518바이트입니다.

단계 **5** 이 패킷 캡처 세션에 포함해야 할 포트를 지정합니다. 여러 포트에서 캡처할 수 있으며 동일한 패킷 캡처 세션 동안 고객 대상 및 애플리케이션 포트 모두에서 캡처할 수 있습니다. 세션에 포함된 각 포트에 별도의 패킷 캡처 파일이 생성됩니다.

참고 패킷 캡처 세션에서 포트를 제거하려면 아래에 나열된 명령에서 **create** 대신 **delete**를 사용합니다.

a) 다음 명령을 사용하여 고객 대상 포트를 추가합니다.
Firepower-chassis /packet-capture/session* # **create** {**phy-port** | **phy-aggr-port**} *port_name*

phy-port의 경우 *port_name* 구문은 **Ethernet**<*slot_id*>/<*port_id*> 또는 **Port-Channel**<*number*>입니다. 브레이크아웃 케이블(phy-aggr-port)의 경우 *port_name* 구문은 **Ethernet**<*slot_id*>/<*port_id*>/<*breakout_port_id*>입니다.

b) 다음 명령을 사용하여 애플리케이션 포트를 추가합니다.
Firepower-chassis /packet-capture/session* # **create app_port** *security_module_slot_id link_name interface_name app_name*

c) 원하는 포트를 모두 추가하려면 필요에 따라 위 단계를 반복합니다.

단계 **6** 캡처 중인 트래픽을 필터링합니다.
패킷 캡처 세션에 포함된 인터페이스에 필터를 적용할 수 있습니다. 필터 생성에 대한 지침은 패킷 캡처의 필터 구성, 206 페이지의 내용을 참조하십시오.

a) 필터를 적용할 인터페이스의 범위를 입력합니다.
Firepower-chassis /packet-capture/session* # **scope** {**phy-port** | **phy-aggr-port**} *port_name*

```
scope phy-port Ethernet<slot_id>/<port_id>
or
scope phy-aggr-port Ethernet<slot_id>/<port_id>/<breakout_port_id>
or
scope <security_module_slot_id> <link_name> <interface_name> <app_name>
```

b) 원하는 필터를 적용합니다.
Firepower-chassis /packet-capture/session/{phy-port|phy-aggr-port|app-port}* # **set** {**source-filter**} *filtername*

참고 포트에서 필터를 제거하려면 **set source-filter ""**를 사용합니다.

c) 추가 필터를 적용하려면 필요에 따라 위 단계를 반복합니다.

단계 **7** 패킷 캡처 세션을 바로 시작하려는 경우 다음 명령을 사용합니다.
Firepower-chassis /packet-capture/session* # **enable**

새로 생성된 패킷 캡처 세션은 기본적으로 비활성화됩니다. 세션을 명시적으로 활성화하면 변경 사항이 커밋될 때 패킷 캡처 세션이 활성화됩니다. 다른 세션이 이미 활성 상태인 경우 세션을 활성화

하면 오류가 발생합니다. 이 세션을 활성화하기 전에 이미 활성 상태인 패킷 캡처 세션을 비활성화해야 합니다.

단계 **8** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /packet-capture/session* # **commit-buffer**

패킷 캡처 세션을 활성화한 경우, 패킷 캡처가 시작됩니다. 세션에서 PCAP 파일을 다운로드하기 전에 캡처를 중지해야 합니다.

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # create session asa1inside
Firepower-chassis packet-capture/session # set session-memory-usage 256
Firepower-chassis packet-capture/session* # create phy-port Ethernet3/1
Firepower-chassis packet-capture/session* # create phy-aggr-port Ethernet2/1/1
Firepower-chassis packet-capture/session* # create app-port 1 link1 Ethernet 1/1 asa
Firepower-chassis packet-capture/session* # create filter interface1vlan100
Firepower-chassis packet-capture/filter* # set ivlan 100
Firepower-chassis packet-capture/filter* # set srcIP 6.6.6.6
Firepower-chassis packet-capture/filter* # set srcPort 80
Firepower-chassis packet-capture/filter* # set destIP 10.10.10.10
Firepower-chassis packet-capture/filter* # set destPort 5050
Firepower-chassis packet-capture/filter* # exit
Firepower-chassis packet-capture/session* # scope phy-port Ethernet3/1
Firepower-chassis packet-capture/session/phy-port* # set src-filter interface1vlan100
Firepower-chassis packet-capture/session/phy-port* # exit
Firepower-chassis packet-capture/session* # scope app-port 1 link1 Ethernet1/1 asa
Firepower-chassis packet-capture/session/app-port* # set src-filter interface1vlan100
Firepower-chassis packet-capture/session/app-port* # exit
Firepower-chassis packet-capture/session* # enable
Firepower-chassis packet-capture/session* # commit-buffer
Firepower-chassis packet-capture/session #
```

패킷 캡처의 필터 구성

필터를 생성하여 패킷 캡처 세션에 포함된 트래픽을 제한할 수 있습니다. 패킷 캡처 세션을 생성하는 동안 특정 필터를 사용해야 하는 인터페이스를 선택할 수 있습니다.

참고 현재 실행 중인 패킷 캡처 세션에 적용된 필터를 수정하거나 삭제하는 경우, 변경 사항은 세션을 비활성화한 다음 다시 활성화할 때까지 적용되지 않습니다.

절차

단계 **1** 캡처 모드를 시작합니다.
Firepower-chassis # **scope packet-capture**

단계 **2** 새 패킷 캡처 필터를 생성합니다.
Firepower-chassis /packet-capture # **create filter** *filter_name*

기존 패킷 캡처 필터를 편집합니다.

Firepower-chassis /packet-capture # **enter filter** *filter_name*

기존 패킷 캡처 필터를 삭제합니다.

Firepower-chassis /packet-capture # **delete filter** *filter_name*

단계 **3** 하나 이상의 필터 속성을 설정하여 필터 세부사항을 지정합니다.
Firepower-chassis /packet-capture/filter* # **set** <*filterprop filterprop_value*

참고 IPv4 또는 IPv6 주소를 사용하여 필터링할 수 있지만 동일한 패킷 캡처 세션에서 두 가지를 모두 필터링할 수는 없습니다.

표 9: 지원되는 필터 속성

ivlan	내부 VLAN ID(포트를 인그레스하는 동안의 패킷 VLAN)
ovlan	외부 VLAN ID(Firepower 4100/9300 섀시에서 추가한 VLAN)
srcip	소스 IP 주소(IPv4)
destip	대상 IP 주소(IPv4)
srcipv6	소스 IP 주소(IPv6)
destipv6	대상 IP 주소(IPv6)
srcport	소스 포트 번호
destport	대상 포트 번호
protocol	IP 프로토콜[10진수 형식의 IANA 정의 프로토콜 값]
ethertype	이더넷 프로토콜 유형[10진수 형식의 IANA 정의 이더넷 프로토콜 유형 값. 예: IPv4 = 2048, IPv6 = 34525, ARP = 2054, SGT = 35081]
srcmac	소스 MAC 주소
destmac	대상 MAC 주소

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # create filter interface1vlan100
Firepower-chassis packet-capture/filter* # set ivlan 100
Firepower-chassis packet-capture/filter* # set srcip 6.6.6.6
Firepower-chassis packet-capture/filter* # set srcport 80
Firepower-chassis packet-capture/filter* # set destip 10.10.10.10
Firepower-chassis packet-capture/filter* # set destport 5050
Firepower-chassis packet-capture/filter* # commit-buffer
```

패킷 캡처 세션 시작 및 중지

절차

단계 **1** 캡처 모드를 시작합니다.
Firepower-chassis # **scope packet-capture**

단계 **2** 시작 또는 중지할 패킷 캡처 세션의 범위를 입력합니다.
Firepower-chassis /packet-capture # **enter session** *session_name*

단계 **3** 다음 명령을 사용하여 패킷 캡처 세션을 시작합니다.
Firepower-chassis /packet-capture/session* # **enable** [append | overwrite]

참고 다른 세션이 실행 중인 동안에는 패킷 캡처 세션을 시작할 수 없습니다.

패킷 캡처 세션이 실행 중인 경우, 개별 PCAP 파일의 파일 크기는 트래픽이 캡처됨에 따라 증가합니다. 버퍼 크기 제한에 도달하면 시스템이 패킷 삭제를 시작하고 Drop Count(삭제 수) 필드가 증가합니다.

단계 **4** 다음 명령을 사용하여 패킷 캡처 세션을 중지합니다.
Firepower-chassis /packet-capture/session* # **disable**

단계 **5** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /packet-capture/session* # **commit-buffer**

패킷 캡처 세션을 활성화한 경우, 세션에 포함된 인터페이스의 PCAP 파일은 트래픽 수집을 시작합니다. 세션이 세션 데이터를 덮어쓰도록 구성된 경우, 기존 PCAP 데이터가 지워집니다. 아닌 경우 데이터가 기존 파일(있는 경우)에 추가됩니다.

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # scope session asa1inside
Firepower-chassis packet-capture/session # enable append
Firepower-chassis packet-capture/session* # commit-buffer
Firepower-chassis packet-capture/session #
```

패킷 캡처 파일 다운로드

네트워크 패킷 분석기를 사용하여 분석할 수 있도록 세션에서 로컬 컴퓨터로 PCAP(패킷 캡처) 파일을 다운로드할 수 있습니다.

PCAP 파일은 `workspace://packet-capture` 디렉토리에 저장되며 다음 명명 규칙을 사용합니다.

`workspace://packet-capture/session-<id>/<session-name>-<interface-name>.pcap`

절차

Firepower 4100/9300 섀시에서 PCAP 파일을 복사하려면 다음을 수행합니다.

참고 패킷 캡처 세션에서 PCAP 파일을 다운로드하기 전에 해당 패킷 캡처 세션을 중지해야 합니다.

a) 로컬 관리에 연결합니다.
Firepower-chassis # **connect localmgmt**

b) PCAP 파일을 복사합니다.
copy *pcap_file copy_destination*

```
Firepower-chassis# connect localmgmt
# copy workspace:/packet-capture/session-1/test-ethernet-1-1-0.pcap
scp://user@10.10.10.1:/workspace/
```

패킷 캡처 세션 삭제

개별 패킷 캡처 세션을 삭제(현재 실행 중이지 않은 경우)하거나 모든 비활성 상태의 패킷 캡처 세션을 삭제할 수 있습니다.

절차

단계 **1** 캡처 모드를 시작합니다.
Firepower-chassis # **scope packet-capture**

단계 **2** 특정 패킷 캡처 세션을 삭제하려면 다음 명령을 사용합니다.
Firepower-chassis /packet-capture # **delete session** *session_name*

단계 **3** 모든 비활성 상태의 패킷 캡처 세션을 삭제하려면 다음 명령을 사용합니다.
Firepower-chassis /packet-capture # **delete-all-sessions**

단계 **4** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.
Firepower-chassis /packet-capture* # **commit-buffer**

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # delete session asa1inside
Firepower-chassis packet-capture* # commit-buffer
Firepower-chassis packet-capture #
```

네트워크 연결성 테스트

시작하기 전에

호스트 이름 또는 IPv4 주소가 있는 네트워크에서 다른 디바이스를 ping하여 기본 네트워크 연결을 테스트하려면 **ping** 명령을 사용합니다. 호스트 이름 또는 IPv6 주소가 있는 네트워크에서 다른 디바이스를 ping하려면 **ping6** 명령을 사용합니다.

호스트 이름 또는 IPv4 주소가 있는 네트워크에서 다른 디바이스에 대한 경로를 추적하려면 **traceroute** 명령을 사용합니다. 호스트 이름 또는 IPv6 주소가 있는 네트워크에서 다른 디바이스에 대한 경로를 추적하려면 **traceroute6** 명령을 사용합니다.

- **ping** 및 **ping6** 명령은 `local-mgmt` 모드에서 사용할 수 있습니다.
- 또한, **ping** 명령은 `module` 모드에서 사용할 수 있습니다.
- **traceroute** 및 **traceroute6** 명령은 `local-mgmt` 모드에서 사용할 수 있습니다.
- 또한, **traceroute** 명령은 `module` 모드에서 사용할 수 있습니다.

절차

단계 **1** 다음 명령 중 하나를 입력하여 `local-mgmt` 또는 `module` 모드에 연결합니다.

- **connect local-mgmt**
- **connect module** *module-ID* {**console** | **telnet**}

예제:

```
FP9300-A# connect local-mgmt
FP9300-A(local-mgmt)#
```

단계 **2** 다음 명령을 사용하여 호스트 이름 또는 IPv4 주소가 있는 네트워크에서 다른 디바이스를 ping하여 기본 네트워크 연결을 테스트합니다.
ping {*hostname* | *IPv4_address*} [**count** *number_packets*] | [**deadline** *seconds*] | [**interval** *seconds*] | [**packet-size** *bytes*]

예제:
이 예에서는 네트워크에 있는 다른 디바이스에 연결하여 ping을 12번 수행하는 방법을 보여줍니다.

```
FP9300-A(local-mgmt)# ping 198.51.100.10 count 12
PING 198.51.100.10 (198.51.100.10) from 203.0.113.5 eth0: 56(84) bytes of data.
64 bytes from 198.51.100.10: icmp_seq=1 ttl=61 time=0.264 ms
64 bytes from 198.51.100.10: icmp_seq=2 ttl=61 time=0.219 ms
64 bytes from 198.51.100.10: icmp_seq=3 ttl=61 time=0.234 ms
64 bytes from 198.51.100.10: icmp_seq=4 ttl=61 time=0.205 ms
64 bytes from 198.51.100.10: icmp_seq=5 ttl=61 time=0.216 ms
64 bytes from 198.51.100.10: icmp_seq=6 ttl=61 time=0.251 ms
64 bytes from 198.51.100.10: icmp_seq=7 ttl=61 time=0.223 ms
64 bytes from 198.51.100.10: icmp_seq=8 ttl=61 time=0.221 ms
64 bytes from 198.51.100.10: icmp_seq=9 ttl=61 time=0.227 ms
```

```
64 bytes from 198.51.100.10: icmp_seq=10 ttl=61 time=0.224 ms
64 bytes from 198.51.100.10: icmp_seq=11 ttl=61 time=0.261 ms
64 bytes from 198.51.100.10: icmp_seq=12 ttl=61 time=0.261 ms

--- 198.51.100.10 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11104ms
rtt min/avg/max/mdev = 51.005/51.062/51.164/0.064 ms

FP9300-A(local-mgmt)#
```

단계 **3** 다음 명령을 사용하여 호스트 이름 또는 IPv4 주소를 사용하여 네트워크에서 다른 디바이스에 대한 경로를 추적합니다.
traceroute {*hostname* | *IPv4_address*}

예제:

```
FP9300-A(local-mgmt)# traceroute 198.51.100.10
traceroute to 198.51.100.10 (198.51.100.10), 30 hops max, 40 byte packets
 1  198.51.100.57 (198.51.100.57)  0.640 ms  0.737 ms  0.686 ms
 2  net1-gw1-13.cisco.com (198.51.100.101)  2.050 ms  2.038 ms  2.028 ms
 3  net1-sec-gw2.cisco.com (198.51.100.201)  0.540 ms  0.591 ms  0.577 ms
 4  net1-fp9300-19.cisco.com (198.51.100.108)  0.336 ms  0.267 ms  0.289 ms

FP9300-A(local-mgmt)#
```

단계 **4** (선택 사항) **exit**을 입력하여 `local-mgmt` 모드를 종료하고 최상위 레벨 모드로 돌아갑니다.

포트 채널 상태 판단

다음 단계를 수행하여 현재 정의된 포트 채널의 상태를 확인할 수 있습니다.

절차

단계 **1** 다음 명령을 입력하여 `/eth-uplink/fabric` 모드를 시작합니다.

- **connect eth-uplink**
- **scope fabric** {**a** | **b**}

예제:

```
FP9300-A# connect eth-uplink
FP9300-A /eth-uplink # scope fabric a
FP9300-A /eth-uplink/fabric #
```

단계 **2** **show port-channel** 명령을 입력하여 각각의 관리 상태 및 작동 상태와 함께 현재 포트 채널 목록을 표시합니다.

예제:

```
FP9300-A /eth-uplink/fabric # show port-channel

Port Channel:
```

```
    Port Channel Id Name             Port Type          Admin
 State Oper State       State Reason
    --------------- ---------------- ------------------ -----
------ ---------------- ------------
    10              Port-channel10   Data               Enabl
ed     Failed           No operational members
    11              Port-channel11   Data               Enabl
ed     Failed           No operational members
    12              Port-channel12   Data               Disab
led    Admin Down       Administratively down
    48              Port-channel48   Cluster            Enabl
ed     Up

FP9300-A /eth-uplink/fabric #
```

단계 **3** 다음 명령을 입력하여 /port-channel 모드를 시작하고 개별 포트 채널 및 포트 정보를 표시합니다.

- **scope port-channel** *ID*

예제:

```
FP9300-A /eth-uplink/fabric/port-channel # top
FP9300-A# connect fxos
Cisco Firepower Extensible Operating System (FX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2017, Cisco Systems, Inc. All rights reserved.

The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license.

    <--- remaining lines removed for brevity --->

FP9300-A(fxos)#
```

단계 **4** **show** 명령을 입력하여 지정된 포트 채널에 대한 상태 정보를 표시합니다.

예제:

```
FP9300-A /eth-uplink/fabric/port-channel # show

Port Channel:
    Port Channel Id Name             Port Type          Admin
 State Oper State       State Reason
    --------------- ---------------- ------------------ -----
------ ---------------- ------------
    10              Port-channel10   Data               Enabl
ed     Failed           No operational members

FP9300-A /eth-uplink/fabric/port-channel #
```

단계 **5** **show member-port** 명령을 입력하여 포트 채널의 멤버 포트에 대한 상태 정보를 표시합니다.

예제:

```
FP9300-A /eth-uplink/fabric/port-channel # show member-port

Member Port:
    Port Name       Membership         Oper State       State Reas
on
    --------------- ------------------ ---------------- ----------
--
    Ethernet2/3     Suspended          Failed           Suspended
    Ethernet2/4     Suspended          Failed           Suspended

FP9300-A /eth-uplink/fabric/port-channel #
```

포트 채널은 논리적 디바이스에 할당될 때까지 나타나지 않습니다. 포트 채널이 논리적 디바이스에서 제거되거나 논리적 디바이스가 삭제된 경우, 포트 채널은 Suspended(일시 중단) 상태로 되돌아갑니다.

단계 **6** 추가 포트 채널 및 LACP 정보를 보려면 `/eth-uplink/fabric/port-channel` 모드를 종료하고 다음 명령을 입력하여 `fxos` 모드를 시작합니다.

- **top**
- **connect fxos**

예제:

단계 **7** **show port-channel summary** 명령을 입력하여 현재 포트 채널에 대한 요약 정보를 표시합니다.

예제:

```
FP9300-A(fxos)# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
--------------------------------------------------------------------
--------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------
--------------------
10    Po10(SD)    Eth      LACP      Eth2/3(s)    Eth2/4(s)

11    Po11(SD)    Eth      LACP      Eth2/1(s)    Eth2/2(s)

12    Po12(SD)    Eth      LACP      Eth1/4(D)    Eth1/5(D)

48    Po48(SU)    Eth      LACP      Eth1/1(P)    Eth1/2(P)
```

추가 **show port-channel** 및 **show lacp** 명령은 `fxos` 모드에서 사용할 수 있습니다. 이러한 명령은 다양한 포트 채널 및 용량, 트래픽, 카운터, 사용량 등 LACP 정보를 표시하는 데 사용할 수 있습니다.

다음에 할 작업

포트 채널 생성 관련 정보는 포트 채널 생성, 142 페이지의 내용을 참조하십시오.

소프트웨어 장애 복구

시작하기 전에

시스템의 성공적인 부팅을 방해하는 소프트웨어 장애가 발생하면 다음 절차에 따라 소프트웨어의 새 버전을 부팅할 수 있습니다. 이 프로세스를 완료하려면 킥스타트 이미지를 TFTP 부팅하고, 새 시스템과 관리자 이미지를 다운로드하고, 새 이미지를 사용하여 부팅해야 합니다.

특정한 FXOS 버전의 복구 이미지는 다음 위치 중 하나의 Cisco.com에서 다운로드할 수 있습니다.

- Firepower 9300 —https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286287252&flowid=77282&softwareid=286287263
- Firepower 4100 Series —https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286305187&flowid=79423&softwareid=286287263

복구 이미지에는 3개의 개별 파일이 들어 있습니다. 예를 들어 아래는 FXOS 2.1.1.64용 최신 복구 이미지입니다.

```
Recovery image (kickstart) for FX-OS 2.1.1.64.
fxos-k9-kickstart.5.0.3.N2.4.11.63.SPA

Recovery image (manager) for FX-OS 2.1.1.64.
fxos-k9-manager.4.1.1.63.SPA

Recovery image (system) for FX-OS 2.1.1.64.
fxos-k9-system.5.0.3.N2.4.11.63.SPA
```

절차

단계 **1** ROMMON에 액세스합니다.

a) 콘솔 포트에 연결합니다.

b) 시스템을 재부팅합니다.
로딩이 시작되고 이 프로세스 동안 카운트다운 타이머가 표시됩니다.

c) ROMMON 모드를 시작하려면 카운트다운 중에 **Escape(**이스케이프**)** 키를 누릅니다.

예제:

```
Cisco System ROMMON, version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015 by Cisco Systems, Inc.
Compiled Sun 01/01/1999 23:59:59:59.99 by user

Current image running: Boot ROM0
Last reset cause: LocalSoft
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address aa:aa:aa:aa:aa:aa

find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.0.00.00.SPA

  bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.0.00.00.SPA

Use BREAK, ESC or CTRL+L to interrupt boot.
use SPACE to begin boot immediately.
Boot interrupted.

rommon 1  >
```

단계 **2** 킥스타트 이미지를 TFTP 부팅합니다.

a) 관리 IP 주소, 관리 넷마스크 및 게이트웨이 IP 주소가 올바르게 설정되었는지 확인합니다. **set** 명령을 사용하여 이러한 값을 볼 수 있습니다. **ping** 명령을 사용하여 TFTP 서버에 대한 연결을 테스트할 수 있습니다.

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "
rommon > address <ip-address>
rommon > netmask <network-mask>
rommon > gateway <default-gateway>
```

b) 킥스타트 이미지를 Firepower 4100/9300 섀시에서 액세스 가능한 TFTP 디렉토리에 복사합니다.

참고 킥스타트 이미지 버전 번호는 번들 버전 번호와 일치하지 않습니다. FXOS 버전과 킥스타트 이미지 간의 매핑을 보여주는 정보는 Cisco.com 소프트웨어 다운로드 페이지에서 확인할 수 있습니다.

c) boot 명령을 사용하여 ROMMON에서 이미지를 부팅합니다.

```
boot tftp://<IP address>/<path to image>
```

참고 Firepower 4100/9300 섀시의 전면 패널에 있는 USB 슬롯에 삽입한 USB 미디어 디바이스를 사용하여 ROMMON에서 킥스타트를 부팅할 수도 있습니다. 시스템이 실행 중일 때 USB 디바이스를 삽입하는 경우 시스템을 리부팅해야 USB 디바이스가 인식됩니다.

이미지가 수신 중임을 나타내는 일련의 # 표시가 나타난 다음 킥스타트 이미지가 로드됩니다.

예제:

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "

rommon 2 > address 10.0.0.2
rommon 3 > netmask 255.255.255.0
rommon 4 > gateway 10.0.0.1
rommon 5 > ping 10.0.0.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)
rommon 6 > ping 192.168.1.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)

rommon 7 > boot tftp://192.168.1.2/fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA
             ADDRESS: 10.0.0.2
             NETMASK: 255.255.255.0
             GATEWAY: 10.0.0.1
              SERVER: 192.168.1.2
               IMAGE: fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA

        TFTP_MACADDR: aa:aa:aa:aa:aa:aa
.....................................................................

Receiving fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA from 192.168.1.2

############################################################################
############################################################################
```

```
###################################################################################

File reception completed.
```

단계 **3** Firepower 4100/9300 섀시에 방금 로드한 킥스타트 이미지와 일치하는 복구 시스템 및 관리자 이미지를 다운로드합니다.

a) 복구 시스템 및 관리자 이미지를 다운로드하려면 관리 IP 주소와 게이트웨이를 설정해야 합니다. USB를 통해 이 이미지를 다운로드할 수 없습니다.

```
switch(boot)# config terminal
switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address <ip address> <netmask>
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway <gateway>
switch(boot)(config)# exit
```

b) 원격 서버에서 부트플래시로 복구 시스템 및 관리자 이미지를 복사합니다.
switch(boot)# **copy** *URL* **bootflash:**

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*/*image_name*
- **scp**://*username@hostname*/*path*/*image_name*
- **sftp**://*username@hostname*/*path*/*image_name*
- **tftp**://*hostname*/*path*/*image_name*

예제**:**

```
switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:

switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
```

c) 이미지를 성공적으로 Firepower 4100/9300 섀시에 복사한 후 nuova-sim-mgmt-nsg.0.1.0.001.bin에서 관리자 이미지로 symlink를 만듭니다. 이 링크는 로드할 관리자 이미지를 로드 메커니즘에 알려줍니다. 어떤 이미지를 로드하려고 하는지와 상관없이 symlink 이름은 항상 nuova-sim-mgmt-nsg.0.1.0.001.bin이어야 합니다.

```
switch(boot)# copy bootflash:<manager-image>
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin
```

예제**:**

```
switch(boot)# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address 10.0.0.2 255.255.255.0
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway 10.0.0.1
switch(boot)(config)# exit
```

```
switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy bootflash:fxos-k9-manager.4.1.1.69.SPA
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin

Copy complete, now saving to disk (please wait)...

switch(boot)#
```

단계 **4** 방금 다운로드한 시스템 이미지를 로드합니다.

```
switch(boot)# load bootflash:<system-image>
```

예제:

```
switch(boot)# load bootflash:fxos-k9-system.5.0.3.N2.4.11.69.SPA
Uncompressing system image: bootflash:/fxos-k9-system.5.0.3.N2.4.11.69.SPA


Manager image digital signature verification successful
...
System is coming up ... Please wait ...


Cisco FPR Series Security Appliance
FP9300-A login:
```

단계 **5** 복구 이미지를 로드한 후 다음 명령을 입력하여 시스템이 이전 이미지 로딩을 시도하지 못하게 합니다.

참고 이 단계는 복구 이미지를 로드한 후 바로 수행해야 합니다.

```
FP9300-A# scope org
FP9300-A /org # scope fw-platform-pack default
FP9300-A /org/fw-platform-pack # set platform-bundle-version ""
Warning: Set platform version to empty will result software/firmware incompatibility issue.
FP9300-A /org/fw-platform-pack* # commit-buffer
```

단계 **6** Firepower 4100/9300 섀시에서 사용할 플랫폼 번들 이미지를 다운로드 및 설치합니다. 자세한 내용은 이미지 관리, 53 페이지를 참조하십시오.

예제:

```
FP9300-A# scope firmware
FP9300-A /firmware # show download-task

Download task:
    File Name Protocol Server          Port       Userid          State
    --------- -------- --------------- ---------- --------------- -----
```

```
    fxos-k9.2.1.1.73.SPA
              Tftp     192.168.1.2            0                  Downloaded
FP9300-A /firmware # show package fxos-k9.2.1.1.73.SPA detail
Firmware Package fxos-k9.2.1.1.73.SPA:
    Version: 2.1(1.73)
    Type: Platform Bundle
    State: Active
Time Stamp: 2012-01-01T07:40:28.000
Build Date: 2017-02-28 13:51:08 UTC
FP9300-A /firmware #
```

손상된 파일 시스템 복구

시작하기 전에

수퍼바이저의 온보드 플래시가 손상되었으며 시스템을 더 이상 성공적으로 시작할 수 없는 경우, 다음 절차를 사용하여 시스템을 복구할 수 있습니다. 이 프로세스를 완료하려면 킥스타트 이미지 TFTP 부팅이 필요하며 플래시를 다시 포맷하고 새 시스템 및 관리자 이미지를 다운로드한 다음 새 이미지를 사용하여 부팅해야 합니다.

참고 이 절차에는 시스템 플래시 재포맷이 포함됩니다. 따라서 시스템이 복구된 후에 시스템을 완전히 다시 구성해야 합니다.

특정한 FXOS 버전의 복구 이미지는 다음 위치 중 하나의 Cisco.com에서 다운로드할 수 있습니다.

- Firepower 9300 —https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286287252&flowid=77282&softwareid=286287263
- Firepower 4100 Series —https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286305187&flowid=79423&softwareid=286287263

복구 이미지에는 3개의 개별 파일이 들어 있습니다. 예를 들어 아래는 FXOS 2.1.1.64용 복구 이미지입니다.

```
Recovery image (kickstart) for FX-OS 2.1.1.64.
fxos-k9-kickstart.5.0.3.N2.4.11.63.SPA

Recovery image (manager) for FX-OS 2.1.1.64.
fxos-k9-manager.4.1.1.63.SPA

Recovery image (system) for FX-OS 2.1.1.64.
fxos-k9-system.5.0.3.N2.4.11.63.SPA
```

절차

단계 **1** ROMMON에 액세스합니다.

a) 콘솔 포트에 연결합니다.

b) 시스템을 재부팅합니다.

시스템이 로딩을 시작하며, 로딩 프로세스 중에 카운트다운 타이머가 표시됩니다.

c) 카운트다운 중에 **Escape** 키를 눌러 ROMMON 모드로 들어갑니다.

예제:

```
Cisco System ROMMON, version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015 by Cisco Systems, Inc.
Compiled Sun 01/01/1999 23:59:59:59.99 by user

Current image running: Boot ROM0
Last reset cause: LocalSoft
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address aa:aa:aa:aa:aa:aa

find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.0.00.00.SPA

  bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.0.00.00.SPA

Use BREAK, ESC or CTRL+L to interrupt boot.
use SPACE to begin boot immediately.
Boot interrupted.

rommon 1  >
```

단계 **2** 킥스타트 이미지를 TFTP 부팅합니다.

a) 관리 IP 주소, 관리 넷마스크 및 게이트웨이 IP 주소가 올바르게 설정되었는지 확인합니다. **set** 명령을 사용하여 이러한 값을 볼 수 있습니다. **ping** 명령을 사용하여 TFTP 서버에 대한 연결을 테스트할 수 있습니다.

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "
rommon > address <ip-address>
rommon > netmask <network-mask>
rommon > gateway <default-gateway>
```

b) 킥스타트 이미지를 Firepower 4100/9300 섀시에서 액세스 가능한 TFTP 디렉토리에 복사합니다.

참고 킥스타트 이미지 버전 번호는 번들 버전 번호와 일치하지 않습니다. Cisco.com 소프트웨어 다운로드 페이지에서 FXOS 버전과 킥스타트 이미지 간 매핑을 보여주는 정보를 찾을 수 있습니다.

c) boot 명령을 사용하여 ROMMON에서 이미지를 부팅합니다.

```
boot tftp://<IP address>/<path to image>
```

참고 USB 미디어 디바이스를 Firepower 4100/9300 섀시의 전면 패널에 있는 USB 슬롯에 삽입하여 ROMMON에서 킥스타트를 부팅할 수도 있습니다. 시스템 실행 중에 USB 디바이스가 삽입된 경우, USB 디바이스를 인식하기 전에 시스템을 재부팅해야 합니다.

이미지가 수신 중임을 나타내는 일련의 # 표시가 나타난 다음 킥스타트 이미지가 로드됩니다.

예제:

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "

rommon 2 > address 10.0.0.2
rommon 3 > netmask 255.255.255.0
rommon 4 > gateway 10.0.0.1
rommon 5 > ping 10.0.0.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)
rommon 6 > ping 192.168.1.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)

rommon 7 > boot tftp://192.168.1.2/fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA
             ADDRESS: 10.0.0.2
             NETMASK: 255.255.255.0
             GATEWAY: 10.0.0.1
              SERVER: 192.168.1.2
               IMAGE: fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA

        TFTP_MACADDR: aa:aa:aa:aa:aa:aa
.....................................................................

Receiving fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA from 192.168.1.2

############################################################################
############################################################################
############################################################################

File reception completed.
```

단계 **3** 킥스타트 이미지가 로드된 후 **init system** 명령을 사용하여 플래시를 다시 포맷합니다. **init system** 명령은 시스템에 다운로드한 모든 소프트웨어 이미지와 시스템의 모든 컨피그레이션을 비롯하여 플래시의 콘텐츠를 지웁니다. 이 명령은 완료하는 데 약 20~30분이 소요됩니다.

예제:

```
switch(boot)# init system

This command is going to erase your startup-config, licenses as well as the contents of
your bootflash:.

Do you want to continue? (y/n)  [n] y

Detected 32GB flash...
Initializing the system
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Initializing startup-config and licenses
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Formatting bootflash:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Formatting SAM partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
```

```
Formatting Workspace partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Formatting Sysdebug  partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
```

단계 **4** 복구 이미지를 Firepower 4100/9300 섀시에 다운로드합니다.

a) 복구 이미지를 다운로드하려면 관리 IP 주소와 게이트웨이를 설정해야 합니다. USB를 통해 이 이미지를 다운로드할 수 없습니다.

```
switch(boot)# config terminal
switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address <ip address> <netmask>
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway <gateway>
switch(boot)(config)# exit
```

b) 원격 서버에서 부트플래시로 세 복구 이미지를 모두 복사합니다.
switch(boot)# **copy** *URL* **bootflash:**

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*/*image_name*
- **scp**://*username@hostname*/*path*/*image_name*
- **sftp**://*username@hostname*/*path*/*image_name*
- **tftp**://*hostname*/*path*/*image_name*

예제:

```
switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  bootflash:

switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:

switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
```

c) 이미지를 성공적으로 Firepower 4100/9300 섀시에 복사한 후 nuova-sim-mgmt-nsg.0.1.0.001.bin에서 관리자 이미지로 symlink를 만듭니다. 이 링크는 로드할 관리자 이미지를 로드 메커니즘에 알려줍니다. 어떤 이미지를 로드하려고 하는지와 상관없이 symlink 이름은 항상 nuova-sim-mgmt-nsg.0.1.0.001.bin이어야 합니다.

```
switch(boot)# copy bootflash:<manager-image>
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin
```

예제:

```
switch(boot)# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(boot)(config)# interface mgmt 0
```

```
switch(boot)(config-if)# ip address 10.0.0.2 255.255.255.0
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway 10.0.0.1
switch(boot)(config)# exit
switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy bootflash:fxos-k9-manager.4.1.1.69.SPA
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin

Copy complete, now saving to disk (please wait)...

switch(boot)#
```

단계 **5** 스위치를 다시 로드합니다.

```
switch(boot)# reload
```

예제:

```
switch(boot)# reload
This command will reboot this supervisor module. (y/n) ? y
[ 1866.310313] Restarting system.

!!  Rommon image verified successfully  !!


Cisco System ROMMON, Version 1.0.11, RELEASE SOFTWARE
Copyright (c) 1994-2016  by Cisco Systems, Inc.
Compiled Wed 11/23/2016 11:23:23.47 by builder
Current image running: Boot ROM1
Last reset cause: ResetRequest
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!
BIOS has been locked !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address: bb:aa:77:aa:aa:bb

autoboot: Can not find autoboot file 'menu.lst.local'
          Or can not find correct boot string !!
```

```
rommon 1 >
```

단계 **6** 킥스타트 및 시스템 이미지에서 부팅합니다.

```
rommon 1 > boot <kickstart-image> <system-image>
```

참고 시스템 이미지를 로드하는 동안 라이선스 관리자 장애 메시지가 표시될 수 있습니다. 이러한 메시지는 안전하게 무시할 수 있습니다.

예제:

```
rommon 1 > dir
Directory of: bootflash:\

  01/01/12  12:33a <DIR>          4,096  .
  01/01/12  12:33a <DIR>          4,096  ..
  01/01/12  12:16a <DIR>         16,384  lost+found
  01/01/12  12:27a           34,333,696  fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  01/01/12  12:29a          330,646,465  fxos-k9-manager.4.1.1.69.SPA
  01/01/12  12:31a          250,643,172  fxos-k9-system.5.0.3.N2.4.11.69.SPA
  01/01/12  12:34a          330,646,465  nuova-sim-mgmt-nsg.0.1.0.001.bin
          4 File(s) 946,269,798 bytes
          3 Dir(s)

rommon 2 > boot fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA fxos-k9-system.5.0.3.N2.4.11.69.SPA
  !!   Kickstart Image verified successfully   !!

Linux version: 2.6.27.47 (security@cisco.com) #1 SMP Thu Nov 17 18:22:00 PST 2016
[    0.000000] Fastboot Memory at 0c100000 of size 201326592
Usage: init 0123456SsQqAaBbCcUu

INIT: version 2.86 booting

POST INIT Starts at Sun Jan  1 00:27:32 UTC 2012
S10mount-ramfs.supnuovaca Mounting /isan 3000m
Mounted /isan
Creating /callhome..
Mounting /callhome..
Creating /callhome done.
Callhome spool file system init done.
Platform is BS or QP MIO: 30
FPGA Version 0x00010500 FPGA Min Version 0x00000600
Checking all filesystems..r.r..r done.
Warning: switch is starting up with default configuration
Checking NVRAM block device ... done
.
FIPS power-on self-test passed
Unpack CMC Application software
Loading system software
Uncompressing system image: bootflash:/fxos-k9-system.5.0.3.N2.4.11.69.SPA


Manager image digital signature verification successful

...

System is coming up ... Please wait ...
nohup: appending output to `nohup.out'

           ---- Basic System Configuration Dialog ----

  This setup utility will guide you through the basic configuration of
  the system. Only minimal configuration including IP connectivity to
  the Fabric interconnect and its clustering mode is performed through these steps.

  Type Ctrl-C at any time to abort configuration and reboot system.
  To back track or make modifications to already entered values,
  complete input till end of section and answer no when prompted
  to apply configuration.
```

```
You have chosen to setup a new Security Appliance. Continue? (y/n):
```

단계 **7** 이미지를 로드한 후, 초기 컨피그레이션 설정을 입력하라는 프롬프트가 표시됩니다. 자세한 내용은 초기 컨피그레이션, 12 페이지를 참조하십시오.

단계 **8** Firepower 4100/9300 섀시에서 사용할 플랫폼 번들 이미지를 다운로드합니다. 플랫폼 번들 이미지 버전은 시스템 복구에 사용한 이미지와 일치해야 합니다. 자세한 내용은 이미지 관리, 53 페이지를 참조하십시오.

예제:

```
FP9300-A# scope firmware
FP9300-A /firmware # show download-task

Download task:
    File Name Protocol Server          Port       Userid          State
    --------- -------- --------------- ---------- --------------- -----
    fxos-k9.2.1.1.73.SPA
              Tftp     192.168.1.2            0                 Downloaded
FP9300-A /firmware # show package fxos-k9.2.1.1.73.SPA detail
Firmware Package fxos-k9.2.1.1.73.SPA:
    Version: 2.1(1.73)
    Type: Platform Bundle
    State: Active
Time Stamp: 2012-01-01T07:40:28.000
Build Date: 2017-02-28 13:51:08 UTC
FP9300-A /firmware #
```

단계 **9** 플랫폼 번들을 성공적으로 다운로드한 후에는, 나중에 시스템을 로드할 때 사용할 수 있도록 킥스타트 및 시스템 이미지를 수동으로 활성화해야 합니다. 실행 중인 버전과 제안된 시작 버전이 일치하기 때문에 이 절차를 사용하여 손상된 파일 시스템을 복구하는 경우 자동 활성화가 지원되지 않습니다.

a) Fabric-interconnect a에 대한 범위를 설정합니다.

```
FP9300-A# scope fabric-interconnect a
```

b) **show version** 명령을 사용하여 실행 중인 커널 버전과 실행 중인 시스템 버전을 확인합니다. 이러한 문자열을 사용하여 이미지를 활성화합니다.

```
FP9300-A /fabric-interconnect # show version
```

c) 다음 명령을 입력하여 이미지를 활성화합니다.

```
FP9300-A /fabric-interconnect # activate firmware
  kernel-version <running_kernel_version> system-version <running_system_version>
commit-buffer
```

참고 서버 상태가 “Disk Failed”(디스크 장애 발생)로 변경될 수 있습니다. 이 메시지에 대해 걱정할 필요가 없으며 이 절차를 계속 진행할 수 있습니다.

d) **show version** 명령을 사용하여 시작 버전이 올바르게 설정되었는지 확인하고 이미지의 활성화 상태를 모니터링합니다.

중요 상태가 “Activating”(활성화)에서 “Ready”(준비)로 변경될 때까지 다음 단계를 진행하지 마십시오.

```
FP9300-A /fabric-interconnect # show version
```

예제:

```
FP9300-A /firmware # top
FP9300-A# scope fabric-interconnect a
FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
    Running-Kern-Vers: 5.0(3)N2(4.11.69)
    Running-Sys-Vers: 5.0(3)N2(4.11.69)
    Package-Vers: 2.1(1.73)
    Startup-Kern-Vers:
    Startup-Sys-Vers:
    Act-Kern-Status: Ready
    Act-Sys-Status: Ready
    Bootloader-Vers:

FP9300-A /fabric-interconnect # activate firmware kernel-version
  5.0(3)N2(4.11.69) system-version 5.0(3)N2(4.11.69)
Warning: When committed this command will reset the end-point
FP9300-A /fabric-interconnect* # commit-buffer
FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
    Running-Kern-Vers: 5.0(3)N2(4.11.69)
    Running-Sys-Vers: 5.0(3)N2(4.11.69)
    Package-Vers: 2.1(1.73)
    Startup-Kern-Vers: 5.0(3)N2(4.11.69)
    Startup-Sys-Vers: 5.0(3)N2(4.11.69)
    Act-Kern-Status: Activating
    Act-Sys-Status: Activating
    Bootloader-Vers:

FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
    Running-Kern-Vers: 5.0(3)N2(4.11.69)
    Running-Sys-Vers: 5.0(3)N2(4.11.69)
    Package-Vers: 2.1(1.73)
    Startup-Kern-Vers: 5.0(3)N2(4.11.69)
    Startup-Sys-Vers: 5.0(3)N2(4.11.69)
    Act-Kern-Status: Ready
    Act-Sys-Status: Ready
    Bootloader-Vers:
```

단계 **10** 시스템을 재부팅합니다.

예제:

```
FP9300-A /fabric-interconnect # top
FP9300-A# scope chassis 1
FP9300-A /chassis # reboot no-prompt
Starting chassis reboot. Monitor progress with the command "show fsm status"
FP9300-A /chassis #
```

마지막으로 Firepower 4100/9300 섀시의 전원을 끈 다음 다시 시작하기 전에 시스템은 각 보안 모듈/엔진의 전원을 끕니다. 이 프로세스에는 약 5~10분이 소요됩니다.

단계 **11** 시스템 상태를 모니터링합니다. 서버 상태는 “Discovery”(검색)에서 “Config”(구성)로 변경된 후 최종적으로 “Ok”(정상)가 되어야 합니다.

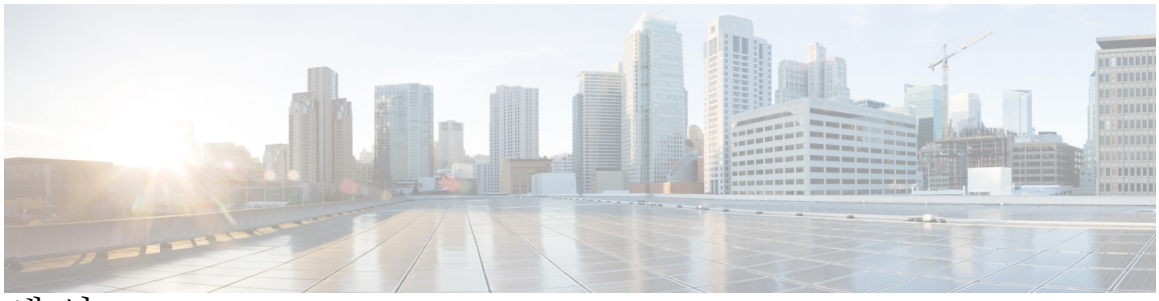
예제:

```
FP9300-A# show server status
Server  Slot Status                        Overall Status        Discovery
------- ---------------------------------- --------------------- ---------
1/1     Equipped                           Discovery             In Progress
1/2     Equipped                           Discovery             In Progress
1/3     Empty

FP9300-A# show server status
```

```
Server  Slot Status                       Overall Status        Discovery
------- --------------------------------- --------------------- ---------
1/1     Equipped                          Config                Complete
1/2     Equipped                          Config                Complete
1/3     Empty

FP9300-A# show server status
Server  Slot Status                       Overall Status        Discovery
------- --------------------------------- --------------------- ---------
1/1     Equipped                          Ok                    Complete
1/2     Equipped                          Ok                    Complete
1/3     Empty
```

Overall Status(전체 상태)가 “Ok”(정상)인 경우, 시스템이 복구된 것입니다. 그래도 보안 어플라이언스를 다시 구성(라이선스 컨피그레이션 포함)하고 논리적 디바이스를 다시 생성해야 합니다. 자세한 내용:

- Firepower 9300 빠른 시작 가이드 —http://www.cisco.com/go/firepower9300-quick
- Firepower 9300 컨피그레이션 가이드—http://www.cisco.com/go/firepower9300-config
- Firepower 4100 Series 빠른 시작 가이드 —http://www.cisco.com/go/firepower4100-quick
- Firepower 4100 Series 컨피그레이션 가이드—http://www.cisco.com/go/firepower4100-config



색 인

H

I

J

K

L

M

P

R

S

T

W