Cisco FXOS CLI 컨피그레이션 가이드, 1.1(1)

초판: 2015년 07월 16일

최종 변경: 2015년 10월 12일

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

텍스트 부품 번호: 온라인 전용

이 설명서의 제품 사양 및 정보는 예고 없이 변경될 수 있습니다. 이 설명서의 모든 설명, 정보 및 권장 사항은 정확한 것으로 간주되지만 이에 대해 명시적이든 묵시적이든 어떠한 보증도 없이 제공됩니다. 모든 제품의 애플리케이션 사용에 대한 책임은 전적으로 사용자에게 있습니다.

동봉된 제품의 소프트웨어 라이센스 및 제한 보증은 제품과 함께 제공되는 정보 패킷에 설명되어 있으며 본 참조 문서에 통합되어 있습니다. 소프트웨어 라이센스 또는 제한 보증을 찾을 수 없는 경우 CISCO 담당자에게 사본을 요청해 주십시오.

Cisco의 TCP 헤더 압축은 UNIX 운영 체제의 UCB 공개 도메인 버전의 일부로서 University of Berkeley(UCB)에서 개발된 프로그램을 적용하여 구현합니다. All rights reserved. Copyright © 1981, Regents of the University of California.

여기에 명시된 다른 모든 보증에도 불구하고 이러한 공급업체의 모든 문서 파일 및 소프트웨어는 모든 결점을 포함하여 "있는 그대로" 제공됩니다. CISCO 및 위에 언급된 모든 공급업체는 상품성, 특정 목적에의 적합성, 타인의 권리 비침해 또는 처리, 사용, 거래 행위로 발생하는 문제에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 모든 종류의 보증을 부인합니다.

CISCO 또는 그 공급자는 이 설명서의 사용 또는 사용할 수 없음으로 인한 모든 파생적, 부수적, 직접, 간접, 특별, 징벌적 또는 기타 모든 손해(영업 이익 손실, 영업 중단, 영업 정보 손실, 또는 그 밖의 금전적 손실로 인한 손해를 포함하되 이에 제한되지 않음)에 대하여 어떠한 경우에도 책임을 지지 않으며, 이는 CISCO 또는 그 공급자가 그와 같은 손해의 가능성을 사전에 알고 있던 경우에도 마찬가지입니다.

이 문서에서 사용된 모든 IP(인터넷 프로토콜) 주소와 전화 번호는 실제 주소와 전화 번호가 아닙니다. 이 문서에 포함된 예제, 명령 표시 출력, 네트워크 토폴로지 다이어그램 및 다른 그림은 이해를 돕기 위한 자료일 뿐이며, 실제 IP 주소나 전화 번호가 사용되었다면 이는 의도하지 않은 우연의 일치입니다.

Cisco 및 Cisco 로고는 미국 및 기타 국가에서 Cisco Systems, Inc. 및/또는 계열사의 상표 또는 등록 상표입니다. Cisco 상표 목록을 확인하려면. http://www.cisco.com/go/trademarks로 이동하십시오. 언급된 타사 상표는 해당 소유주의 재산입니다. 파트너라는 용어의 사용이 Cisco와 다른 업체 사이의 제휴 관계를 의미하는 것은 아닙니다. (1110R)

목 차

1 장

Firepower Security Appliance 소개

Firepower Security Appliance 정보

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

FXOS 섀시에서 제공하는 기능은 다음과 같습니다.

- 모듈형 섀시 기반 보안 시스템 - 고성능의 유연한 인풋/아웃풋 컨피그레이션 및 확장성을 제공합니다.
- Firepower Chassis Manager - 그래픽 사용자 인터페이스는 현재 섀시 상태 및 섀시 기능의 간소화된 컨피그레이션을 간단하게 시각적으로 표시합니다.
- FXOS CLI - 기능 구성, 섀시 상태 모니터링 및 고급 문제 해결 기능 액세스를 위해 명령 기반 인터페이스를 제공합니다.
- FXOS REST API - 사용자가 섀시를 프로그래밍 방식으로 구성 및 관리할 수 있습니다.

2 장

CLI(Command Line Interface) 개요

관리 객체

Firepower eXtensible 운영 체제은 관리 객체 모델을 사용하며, 여기서 관리 객체는 관리 가능한 물리적 또는 논리적 엔터티를 추상화한 것입니다. 예를 들어, 섀시, 보안 모듈, 네트워크 모듈, 포트 및 프로세서는 관리 객체로 표시된 물리적 엔터티이며 라이센스, 사용자 역할 및 플랫폼 정책은 관리 객체로 표시된 논리적 엔터티입니다.

관리 객체에는 구성 가능한 연결된 속성이 하나 이상 있을 수 있습니다.

명령 모드

CLI에는 명령 모드가 계층 구조로 구성되어 있으며, EXEC 모드는 계층 구조에서 최고 수준의 모드입니다. 상위 수준의 모드는 하위 수준의 모드로 나눠집니다. **create**, **enter** 및 **scope** 명령을 사용하여 상위 수준의 모드에서 다음으로 낮은 수준의 모드로 이동하고 **exit** 명령을 사용하여 모드 계층 구조의 한 수준 위로 이동합니다. 또한 **top** 명령을 사용하여 모드 계층 구조에서 최상위 수준으로 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** 명령은 관리 객체를 생성하지 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 모든 계층 구조의 전체 경로를 보여줍니다. 이 경로는 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 사용된 명령 및 각 모드와 연결된 CLI 프롬프트가 나와 있습니다.

표 ***1:*** 기본 명령 모드 및 프롬프트

모드 이름	액세스하는 데 사용된 명령	모드 프롬프트
실행	모든 모드의 **top** 명령	#
어댑터	EXEC 모드의 **scope adapter** 명령	/adapter #
케이블링	EXEC 모드의 **scope cabling** 명령	/cabling #
chassis	EXEC 모드의 **scope chassis** 명령	/chassis #
이더넷 서버	EXEC 모드의 **scope eth-server** 명령	/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	/host-eth-if #
라이센스	EXEC 모드의 **scope license** 명령	/license #

모드 이름	액세스하는 데 사용된 명령	모드 프롬프트
모니터링	EXEC 모드의 **scope monitoring** 명령	/monitoring #
구성	EXEC 모드의 **scope org** 명령	/org #
보안	EXEC 모드의 **scope security** 명령	/security #
server	EXEC 모드의 **scope server** 명령	/server #
서비스 프로파일	EXEC 모드의 **scope service-profile** 명령	/service-profile #
ssa	EXEC 모드의 **scope ssa** 명령	/ssa #
system	EXEC 모드의 **scope system** 명령	/system #
가상 HBA	EXEC 모드의 **scope vhba** 명령	/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 일치하는 **delete** *object* 및 **enter** *object* 명령이 있습니다.

사용자가 인스턴스화한 객체 관리 시 이러한 명령의 동작은 다음 표에 설명된 대로 객체가 존재하는지 여부에 따라 달라집니다.

표 *2:* 객체가 없는 경우의 일반적인 동작

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 동작

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 탭 키를 사용하여 명령을 완성할 수 있습니다. 명령 이름의 일부를 입력하고 탭 키를 누르면 전체 명령이 표시되거나 다른 키워드를 선택해야 하거나 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 키는 저장된 이전 명령으로 이동하고 아래쪽 화살표 키는 저장된 다음 명령으로 이동합니다. 저장된 마지막 명령에 도달하여 아래쪽 화살표 키를 누르면 아무 명령도 실행되지 않습니다.

단순히 저장된 명령을 하나씩 살펴보고 원하는 명령을 불러온 다음 Enter를 눌러 저장된 모든 명령을 다시 입력할 수 있습니다. 명령어는 사용자가 수동으로 입력한 것처럼 입력됩니다. Enter를 누르기 전에 명령어를 불러 변경할 수도 있습니다.

보류 중인 명령 커밋, 삭제 및 보기

CLI에서 컨피그레이션 명령어를 입력하면 **commit-buffer** 명령을 입력할 때까지 해당 명령이 적용되지 않습니다. 커밋될 때까지 컨피그레이션 명령어는 보류 상태이며 **discard-buffer** 명령을 입력하여 삭제할 수 있습니다.

여러 명령 모드에서 보류 중인 변경 사항을 누적하고 단일 **commit-buffer** 명령으로 함께 적용할 수 있습니다. 모든 명령 모드에서 **show configuration pending** 명령을 입력하여 보류 중인 명령을 확인할 수 있습니다.

참고 여러 명령을 함께 커밋하는 것은 원자성 작업이 아닙니다. 명령에 실패하는 경우에도 성공적인 명령이 적용됩니다. 실패한 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 동안 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 #
```

CLI를 위한 온라인 도움말

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

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

CLI 세션 제한

Firepower eXtensible 운영 체제은 한 번에 활성화할 수 있는 CLI 세션의 수를 총 32개로 제한합니다. 이 값은 구성할 수 없습니다.

3 장

시작하기

작업 흐름

다음 절차에서는 FXOS 섀시 구성 시 완료해야 하는 기본 작업을 보여줍니다.

절차

단계 **1** FXOS 섀시 하드웨어를 구성합니다(Cisco Firepower Security Appliance 하드웨어 설치 설명서 참고).
단계 **2** 초기 컨피그레이션을 완료합니다(초기 컨피그레이션, 10 페이지 참고).
단계 **3** 날짜 및 시간을 설정합니다(날짜 및 시간 설정, 47 페이지 참고).
단계 **4** DNS 서버를 구성합니다(DNS 서버 구성, 72 페이지 참고).
단계 **5** 제품 라이센스를 등록합니다(라이센스 관리, 15 페이지 참고).
단계 **6** 사용자를 구성합니다(사용자 관리, 23 페이지 참고).
단계 **7** 필요한 경우 소프트웨어 업데이트를 수행합니다(이미지 관리, 37 페이지 참고).
단계 **8** 추가 플랫폼 설정을 구성합니다(플랫폼 설정, 45 페이지 참고).
단계 **9** 인터페이스를 구성합니다(인터페이스 관리, 75 페이지 참고).
단계 **10** 논리적 디바이스를 생성합니다(논리적 디바이스, 79 페이지 참고).

초기 컨피그레이션

Firepower Chassis Manager 또는 FXOS CLI를 사용하여 시스템을 구성하고 관리하려면 먼저 콘솔 포트에 액세스하는 데 사용되는 FXOS CLI를 사용하여 초기 컨피그레이션 작업 중 일부를 수행해야 합니다. FXOS 섀시에 FXOS CLI를 사용하여 처음에 액세스할 때 시스템을 구성하는 데 사용할 수 있는 설정 마법사가 나타납니다.

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

FXOS 섀시의 단일 관리 포트에는 하나의 IPv4 주소, 게이트웨이 및 서브넷 마스크 또는 하나의 IPv6 주소, 게이트웨이 및 네트워크 접두사만 지정해야 합니다. 관리 포트 IP 주소에 대해 IPv4 또는 IPv6 주소를 구성할 수 있습니다.

시작하기 전에

1 FXOS 섀시에서 다음의 물리적 연결을 확인합니다.

- 콘솔 포트는 컴퓨터 터미널 또는 콘솔 서버에 물리적으로 연결됩니다.
- 1Gbps 이더넷 관리 포트는 외부 허브, 스위치 또는 라우터에 연결됩니다.

자세한 내용은 Cisco Firepower Security Appliance 하드웨어 설치 설명서를 참고하십시오.

2 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 매개변수가 다음과 같은지 확인하십시오.

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

절차

단계 **1** 콘솔 포트에 연결합니다.

단계 **2** FXOS 섀시의 전원을 켭니다.
FXOS 섀시가 부팅할 때 자체 전원 테스트 메시지를 확인할 수 있습니다.

단계 **3** 구성되지 않은 시스템을 부팅할 경우, 설정 마법사가 시스템을 구성하는 데 필요한 다음 정보에 대해 묻는 메시지를 표시합니다.

- 설정 모드(전체 시스템 백업 또는 초기 설정에서 복원)
- 강력한 비밀번호 시행 정책(강력한 비밀번호 지침에 대해서는 사용자 계정, 23 페이지 참고)
- 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(**예**)**를 입력하여 설정을 저장하고 적용하거나 **no(**아니요**)**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**를 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 전원이 연결된 터미널을 사용하여 FXOS CLI에 연결할 수 있습니다. 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 매개변수가 다음과 같은지 확인하십시오.

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

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

SSH, 텔넷 또는 Putty를 사용하여 로그인하려면 다음 구문 예시 중에서 하나를 사용합니다.

참고 SSH 로그인은 대/소문자를 구분합니다.

SSH를 사용하는 Linux 터미널에서 다음 구문을 사용합니다.

- **ssh ucs-***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
```

- **ssh ucs-***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, 51 페이지을 참고하십시오.

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

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

- **telnet ucs-**{*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.

- 다음으로 로그인: **ucs-***auth-domain\username*

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

참고 기본 인증이 로컬로 설정되어 있고 콘솔 인증이 LDAP로 설정된 경우, ucs-local\admin을 사용하는 Putty 클라이언트에서 Fabric Interconnect에 로그인할 수 있으며 이때 admin은 로컬 계정의 이름입니다.

4 장

라이센스 관리

Cisco Smart Software Licensing은 라이센스 풀을 중앙에서 구입하고 관리하게 해줍니다. 각 유닛의 라이센스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. Smart Software Licensing은 라이센스 사용량과 수요를 한 번에 확인하게 해줍니다.

스마트 소프트웨어 라이센싱 소개

이 섹션에서는 스마트 소프트웨어 라이센싱이 어떻게 적용되는지 설명합니다.

FXOS 섀시의 애플리케이션을 위한 스마트 소프트웨어 라이센싱

애플리케이션(FXOS 섀시)의 경우, 스마트 소프트웨어 라이센싱 컨피그레이션은 FXOS 섀시 수퍼바이저와 애플리케이션으로 나뉩니다.

- FXOS 섀시 - 수퍼바이저의 모든 스마트 소프트웨어 라이센싱 인프라를 구성하며 여기에는 라이센스 기관과 통신하는 데 필요한 매개변수가 포함됩니다. FXOS 섀시 자체를 작동하기 위한 라이센스는 필요하지 않습니다.
- 애플리케이션 - 애플리케이션의 모든 라이센스 엔타이틀먼트를 구성합니다.

Smart Software Manager 및 계정

디바이스 라이센스를 1개 이상 구매한 경우, Cisco Smart Software Manager에서 라이센스를 관리하십시오.

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

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

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

기본적으로 마스터 계정의 기본 가상 계정에 라이센스가 지정됩니다. 어카운트 관리자로서, 선택적으로 추가 가상 어카운트를 생성할 수 있습니다. 예를 들어, 지역, 부서 또는 자회사에 대해 어카운트를 만들 수 있습니다. 여러 가상 어카운트를 활용하면 많은 라이센스 및 디바이스를 보다 쉽게 관리할 수 있습니다.

가상 계정으로 관리되는 라이센스 및 디바이스

라이센스 및 디바이스는 가상 계정별로 관리됩니다. 가상 계정의 디바이스에서만 해당 계정에 할당된 라이센스를 사용할 수 있습니다. 추가 라이센스가 필요한 경우, 다른 가상 어카운트에서 사용하지 않는 라이센스를 전송할 수 있습니다. 또한 가상 계정 간에 디바이스를 이전할 수도 있습니다.

FXOS 섀시만 디바이스로 등록하는 반면 섀시의 애플리케이션은 고유한 라이센스를 요청합니다. 예를 들어, 3개의 보안 모듈이 있는 Firepower 9300 섀시의 경우, 섀시는 1개의 디바이스로 간주되지만 모듈은 3개의 개별 라이센스를 사용합니다.

디바이스 등록 및 토큰

각 가상 어카운트에 대해, 등록 토큰을 만들 수 있습니다. 이 토큰은 기본적으로 30일 동안 유효합니다. 각 디바이스를 구축하거나 기존 디바이스를 등록할 때 이 토큰 ID와 엔타이틀먼트 레벨을 입력합니다. 기존 토큰이 만료되는 경우 새 토큰을 만들 수 있습니다.

참고 디바이스 등록은 FXOS 섀시 수퍼바이저(보안 모듈 아님)에서 구성됩니다.

구축 이후 시작할 때 또는 기존 디바이스에서 이 매개변수를 수동으로 구성한 후에 디바이스가 Cisco 라이센스 기관에 등록됩니다. 디바이스를 토큰과 함께 등록하면 라이센스 기관은 디바이스와 라이센스 기관 간의 통신을 위해 ID 인증서를 발급합니다. 이 인증서는 6개월 마다 갱신되지만 1년 동안 유효합니다.

라이센스 기관과의 주기적인 통신

디바이스는 30일마다 라이센스 기관과 통신합니다. Smart Software Manager에서 변경할 경우 변경 사항이 즉시 적용되도록 디바이스에서 권한 부여를 새로 고칠 수 있습니다. 또는 디바이스에서 예정대로 통신할 때까지 기다릴 수 있습니다.

선택적으로 HTTP 프록시를 구성할 수 있습니다. 최소 90일마다 디바이스가 직접 또는 HTTP 프록시를 통해 인터넷에 연결되어야 합니다. 일반 라이센스 통신은 30일마다 이루어지지만, 유예 기간이 있으므로 디바이스는 최대 90일간 콜 홈 없이 작동할 수 있습니다. 90일이 지나기 전에 Licensing Authority에 접속해야 합니다.

참고 오프라인 라이센싱은 지원되지 않습니다.

규정 위반 상태

디바이스는 다음과 같은 상황에서 규정 위반이 될 수 있습니다.

- 과다 사용 - 디바이스에서 사용 불가한 라이센스를 사용할 경우
- 라이센스 만료 - 시간 기반 라이센스가 만료하는 경우
- 통신 부재 - 디바이스에서 권한 재부여를 위해 라이센싱 기관에 연결하지 못한 경우

권한 재부여 시도 90일 이후에 디바이스는 애플리케이션에 따라 일부분이 제한됩니다.

Smart Call Home 인프라

기본적으로, Smart Call Home 프로필은 라이센싱 기관의 URL을 지정하는 컨피그레이션에 있습니다. 이 프로필을 제거할 수 없습니다. License 프로필의 유일한 컨피그레이션 옵션은 라이센스 기관의 대상 주소 URL입니다. Cisco TAC에서 지시하지 않는 한 라이센스 기관 URL을 변경해서는 안 됩니다.

스마트 소프트웨어 라이센싱을 위해 Smart Call Home을 비활성화할 수 없습니다.

스마트 소프트웨어 라이센싱 사전 요구 사항

- Cisco Smart Software Manager에서 마스터 어카운트를 만듭니다.

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

  아직 계정이 없는 경우 새 계정 설정 링크를 클릭합니다. Smart Software Manager를 활용하면 조직에서 사용할 마스터 어카운트를 만들 수 있습니다.

- Cisco Software Central에서 라이센스를 하나 이상 구매합니다.

- 디바이스에서 Licensing Authority와 통신할 수 있도록 디바이스가 인터넷 액세스 또는 HTTP 프록시 액세스를 보장합니다. 오프라인 라이센싱은 지원되지 않습니다.
- 디바이스에서 라이센싱 기관 서버의 이름을 확인할 수 있도록 DNS 서버를 구성합니다. DNS 서버 구성, 72 페이지을 참고하십시오.
- 디바이스의 클록을 설정합니다. 날짜 및 시간 설정, 47 페이지를 참고하십시오.

스마트 소프트웨어 라이센싱의 기본값

FXOS 섀시 기본 컨피그레이션은 Smart Call Home 프로필인 "SLProf"를 포함하며, 여기에서 라이센스 기관의 URL을 지정합니다.

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

스마트 소프트웨어 라이센싱 구성

Cisco 라이센스 기관과 통신하기 위해 필요에 따라 HTTP 프록시를 구성할 수 있습니다. 라이센스 기관에 등록하려면 스마트 소프트웨어 라이센스 계정에서 얻은 FXOS 섀시에 등록 토큰 ID를 입력해야 합니다.

절차

단계 **1** (선택 사항) HTTP 프록시 구성, 18 페이지.

단계 **2** 라이센스 기관에 Firepower Security Appliance 등록, 19 페이지.

(선택 사항) HTTP 프록시 구성

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

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*ip_address*

예제:

```
set http-proxy-server-url 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

라이센스 기관에 **Firepower Security Appliance** 등록

FXOS 섀시를 등록하면 라이센스 기관은 FXOS 섀시와 라이센스 기관 간 통신을 위한 ID 인증서를 발급합니다. 또한 FXOS 섀시를 적절한 가상 계정에 할당합니다. 일반적으로 이 절차는 1회 수행됩니다. 그러나 예를 들어 통신 문제 때문에 ID 인증서가 만료되면 나중에 FXOS 섀시를 다시 등록해야 할 수 있습니다.

절차

단계 **1** Smart Software Manager에서, 이 FXOS 섀시를 추가할 가상 계정에 대한 등록 토큰을 요청 및 복사합니다.

단계 **2** FXOS 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

FXOS 섀시의 등록을 취소하면 계정에서 해당 디바이스가 제거됩니다. 디바이스의 모든 라이센스 엔타이틀먼트 및 인증서가 제거됩니다. 새 FXOS 섀시의 라이센스를 위해 공간을 비워두려면 등록을 취소할 수 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, 라이센스 자격은 30일마다 갱신됩니다. 예를 들어 인터넷 액세스 기간이 제한된 경우 또는 Smart Software Manager에서 라이센스를 변경한 경우, 이러한 항목 중 하나에 대한 등록을 수동으로 갱신할 수 있습니다.

스마트 소프트웨어 라이센싱 모니터링

라이센스 상태를 보려면 다음 명령을 참조하십시오.

- **show license all**

  스마트 라이센싱 상태, 스마트 에이전트 버전, UDI 정보, 스마트 에이전트 상태, 글로벌 규정 준수 상태, 엔타이틀먼트 상태, 라이센싱 인증서 정보 및 스마트 에이전트 작업 일정을 표시합니다.

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

스마트 소프트웨어 라이센싱 기록

기능 이름	플랫폼 릴리스	설명
FXOS 섀시용 Cisco Smart Software Licensing	1.1(1)	Smart Software Licensing은 라이센스 풀을 구입하고 관리하게 해줍니다. 스마트 라이센스는 특정 일련 번호에 묶여 있지 않습니다. 각 유닛의 라이센스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. Smart Software Licensing은 라이센스 사용량과 수요를 한 번에 확인하게 해줍니다. 스마트 소프트웨어 라이센싱 컨피그레이션은 FXOS 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 계정

사용자 계정은 시스템에 액세스하는 데 사용됩니다. 최대 48개의 로컬 사용자 계정을 구성할 수 있습니다. 각 사용자 계정에는 고유한 사용자 이름 및 비밀번호가 있어야 합니다.

관리자 계정

관리자 계정은 기본 사용자 계정이며 수정 또는 삭제할 수 없습니다. 이 계정은 시스템 관리자 또는 수퍼바이저 계정이며 전체 권한을 가집니다. 관리자 계정에 할당된 기본 비밀번호가 없습니다. 초기 시스템을 설정하는 동안 비밀번호를 선택해야 합니다.

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

로컬에서 인증된 사용자 계정

로컬에서 인증된 사용자 계정은 섀시를 통해 직접 인증되며 관리자 또는 AAA 권한을 보유한 사용자에 의해 활성화 또는 비활성화될 수 있습니다. 로컬 사용자 계정을 비활성화한 경우, 사용자는 로그인할 수 없습니다. 비활성화된 로컬 사용자 계정에 대한 컨피그레이션 세부사항은 데이터베이스에서 삭제되지 않습니다. 비활성화된 로컬 사용자 계정을 다시 활성화하는 경우, 계정이 사용자 이름 및 비밀번호를 포함하여 기존 컨피그레이션으로 다시 활성화됩니다.

원격으로 인증된 사용자 계정

원격으로 인증된 사용자 계정은 LDAP, RADIUS 또는 TACACS+를 통해 인증되는 사용자 계정입니다.

사용자가 로컬 사용자 계정과 원격 사용자 계정을 동시에 유지 관리할 경우, 로컬 사용자 계정에 정의된 역할은 원격 사용자 계정에서 유지 관리되는 역할을 재정의합니다.

사용자 계정 만료

사용자 계정은 미리 정의된 시간에 만료되도록 구성할 수 있습니다. 만료 시간에 도달하면 사용자 계정은 비활성화됩니다.

기본적으로, 사용자 계정은 만료되지 않습니다.

만료일이 있는 사용자 계정을 구성한 후에는 이 계정을 만료되지 않도록 재구성할 수 없습니다. 단, 사용 가능한 최신 만료일의 계정을 구성할 수 있습니다.

사용자 이름 지침

사용자 이름은 Firepower Chassis Manager 및 FXOS CLI에 대한 로그인 ID로도 사용됩니다. 사용자 계정에 로그인 ID를 할당할 때 다음 지침 및 제한 사항을 고려하십시오.

- 로그인 ID는 1자에서 32자로 다음을 포함할 수 있습니다.
  - 알파벳 문자
  - 숫자
  - _(밑줄)
  - -(대시)
  - . (점)
- 로그인 ID는 고유해야 합니다.
- 로그인 ID는 알파벳 문자로 시작해야 합니다. 숫자 또는 밑줄과 같은 특수 문자로 시작할 수 없습니다.
- 로그인 ID는 대/소문자를 구분합니다.
- 모두 숫자인 로그인 ID를 생성할 수 없습니다.

- 사용자 계정을 생성한 후, 로그인 ID를 변경할 수 없습니다. 사용자 계정을 삭제하고 새로 생성해야 합니다.

비밀번호 지침

비밀번호는 각각의 로컬에서 인증된 사용자 계정에 필요합니다. 관리자 또는 AAA 권한이 있는 사용자는 사용자 비밀번호에 대한 비밀번호 보안 수준 확인을 수행하도록 시스템을 구성할 수 있습니다. 비밀번호 보안 수준 확인이 활성화된 경우 각 사용자에게는 강력한 비밀번호가 있어야 합니다.

각 사용자별로 강력한 비밀번호를 사용할 것을 권장합니다. 로컬에서 인증된 사용자를 위해 비밀번호 보안 수준 확인을 활성화한 경우, Firepower eXtensible 운영 체제는 다음 요건을 충족하지 않는 비밀번호를 거부합니다.

- 최소 8자 이상, 최대 80자를 포함해야 합니다.
- 다음 중 최소 3가지 이상을 포함해야 합니다.
  - 소문자
  - 대문자
  - 숫자
  - 특수 문자
- 3번 이상 연속하여 반복되는 문자(예: aaabbb)를 포함할 수 없습니다.
- 3개의 연속 숫자(예: password123)를 포함할 수 없습니다.
- 사용자 이름 또는 사용자 이름을 반대로 한 이름과 동일하지 않아야 합니다.
- 비밀번호 딕셔너리 검사를 통과해야 합니다. 예를 들어, 비밀번호는 표준 사전 단어에 기반을 둘 수 없습니다.
- 다음 기호는 포함할 수 없습니다. 예: $(달러 기호), ? (물음표) 및 =(등호)
- 로컬 사용자 및 관리자 계정이 비어 있지 않아야 합니다.

기본 사용자 역할

시스템에는 다음과 같은 기본 사용자 역할이 포함되어 있습니다.

관리자

전체 시스템에 모든 읽기 및 쓰기 액세스가 가능합니다. 기본 관리자 계정이 기본적으로 이 역할에 할당되며 변경할 수 없습니다.

읽기 전용

시스템 상태를 수정할 권한이 없으며 시스템 컨피그레이션에 읽기 전용으로 액세스합니다.

로컬에서 인증된 사용자용 비밀번호 프로필

비밀번호 프로필에는 로컬에서 인증된 모든 사용자에 대한 비밀번호 기록 및 비밀번호 변경 간격 속성이 포함되어 있습니다. 각각의 로컬에서 인증된 사용자에 대해 다른 비밀번호 프로필을 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 기록 수 임계값에 도달할 경우 가장 오래된 비밀번호만 재사용될 수 있도록 최신 비밀번호가 먼저 저장됩니다.

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

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

필요한 경우, 로컬에서 인증된 사용자에 대한 비밀번호 기록 수를 지우고 이전 비밀번호 재사용을 활성화할 수 있습니다.

비밀번호 변경 간격

비밀번호 변경 간격을 사용하면 로컬에서 인증된 사용자가 지정된 시간 이내에 변경할 수 있는 비밀번호 변경 수를 제한할 수 있습니다. 다음 표에서는 비밀번호 변경 간격에 대한 2가지 컨피그레이션 옵션을 설명합니다.

간격 컨피그레이션	설명	예
비밀번호 변경 허용 안 됨	이 옵션을 사용하면 비밀번호 변경 이후 지정된 시간 동안에는 로컬에서 인증된 사용자의 비밀번호를 변경할 수 없습니다. 1시간에서 745시간 사이로 변경 안 함 간격을 지정할 수 있습니다. 기본적으로, 변경 안 함 간격은 24시간입니다.	예를 들어, 로컬에서 인증된 사용자가 비밀번호를 변경한 후 48시간 이내에 비밀번호가 변경되는 것을 방지하려면 다음을 설정하십시오. • 간격 동안 변경을 비활성화로 설정 • 변경 안 함 간격을 48시간으로 설정

간격 컨피그레이션	설명	예
변경 간격 이내에 허용되는 비밀번호 변경	이 옵션은 로컬에서 인증된 사용자의 비밀번호를 미리 정의한 간격 이내에 변경할 수 있는 최대 횟수를 지정합니다. 변경 간격을 1시간에서 745시간 사이로 지정하고 비밀번호 변경 최대 횟수를 0에서 10 사이로 지정할 수 있습니다. 기본적으로, 로컬에서 인증된 사용자에게는 48시간 간격 이내에 최대 2회의 비밀번호 변경이 허용됩니다.	예를 들어, 로컬에서 인증된 사용자가 비밀번호를 변경한 후 24시간 이내에 비밀번호를 최대 한 번 변경하도록 허용하려면 다음을 설정하십시오. • 간격 동안 변경을 활성화로 설정 • 변경 횟수를 1로 설정 • 변경 간격을 24로 설정

기본 인증 서비스 선택

절차

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

단계 **2** 기본 권한 부여 보안 모드를 입력합니다.
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) 있는 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 새로 고치기 요청 사이에 허용되는 최대 시간을 지정합니다.
Firepower-chassis /security/default-auth # **set refresh-period***seconds*

60~172800의 정수를 지정합니다. 기본값은 600초입니다.

이 시간 제한을 초과할 경우, Firepower eXtensible 운영 체제에서는 웹 세션을 비활성화 상태로 간주하지만 세션을 종료하지는 않습니다.

단계 **6** (선택 사항) Firepower eXtensible 운영 체제에서 웹 세션이 종료되었다고 간주하기 전에 마지막 새로 고치기 요청 이후에 경과한 최대 시간을 지정합니다.
Firepower-chassis /security/default-auth # **set session-timeout***seconds*

60~172800의 정수를 지정합니다. 기본값은 7200초입니다.

참고 RADIUS 또는 TACACS+ 영역에 대해 2가지 계수 인증을 설정한 경우, 원격 사용자가 너무 자주 재인증할 필요가 없도록 **session-refresh** 및 **session-timeout** 간격을 늘리는 것을 고려하십시오.

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

참고 2가지 계수 인증은 RADIUS 및 TACACS+ 영역에만 적용됩니다.

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

다음의 예에서는 기본 인증을 RADIUS에 설정하고 기본 인증 제공자 그룹을 provider1에 설정하고 2가지 계수 인증을 활성화하고 새로 고치기 간격을 7200초(2시간)로 설정하며 세션 시간 초과 간격을 28800초(8시간)로 설정하고 2가지 계수 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 7200
Firepower-chassis /security/default-auth* # set session-timeout 28800
Firepower-chassis /security/default-auth* # commit-buffer
Firepower-chassis /security/default-auth #
```

원격 사용자의 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.

원격 사용자의 역할 정책을 다음 방법으로 구성할 수 있습니다.

assign-default-role

사용자가 로그인을 시도하고 원격 인증 제공자가 인증 정보와 함께 사용자 역할을 제공하지 않는 경우, 사용자는 읽기 전용 사용자 역할에 로그인할 수 있습니다.

이는 기본 동작입니다.

no-login

사용자가 로그인을 시도하고 원격 인증 제공자가 인증 정보와 함께 사용자 역할을 제공하지 않는 경우, 액세스가 거부됩니다.

절차

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

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**

다음의 예는 원격 사용자의 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 #
```

로컬에서 인증된 사용자용 비밀번호 보안 수준 확인 활성화

비밀번호 보안 수준 확인이 활성화된 경우에는 Firepower eXtensible 운영 체제에서 사용자가 강력한 비밀번호 지침을 충족하지 않는 비밀번호를 선택하도록 허용하지 않습니다(비밀번호 지침, 25 페이지 참고).

절차

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

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

다음 예에서는 비밀번호 보안 수준 확인을 활성화합니다.

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

변경 간격 동안 비밀번호 변경 최대 횟수 구성

절차

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

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

단계 **3** 로컬에서 인증된 사용자가 지정된 시간 이내에 변경할 수 있는 비밀번호 변경 수를 제한합니다.
Firepower-chassis /security/password-profile # **set change-during-interval enable**

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

이 값은 0~10의 모든 수가 가능합니다.

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

이 값은 1~745(시간)의 모든 수가 가능합니다.

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

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

다음의 예는 간격 옵션 동안의 변경을 활성화하고 변경 횟수를 5로 설정하고 변경 간격을 72시간으로 설정하며 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 #
```

비밀번호에 대해 변경 안 함 간격 구성

절차

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

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

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

단계 **4** 로컬에서 인증된 사용자가 새로 생성된 비밀번호를 변경하기 전에 대기해야 하는 최소 시간을 지정합니다.
Firepower-chassis /security/password-profile # **set no-change-interval** *min-num-hours*

이 값은 1~745(시간)의 모든 수가 가능합니다.

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

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

다음의 예는 간격 옵션 동안 변경을 비활성화하고 변경 안 함 간격을 72시간으로 설정하며 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 입력합니다.
Firepower-chassis # **scopesecurity**

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

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

이 값은 0~15의 모든 수가 가능합니다.

기본적으로, **History Count(**기록 수**)** 필드가 0으로 설정되어 있어 기록 수가 비활성화되고 사용자가 언제든지 이전에 사용한 비밀번호를 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 계정 생성

절차

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

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

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

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

비밀번호를 입력합니다. *password*

비밀번호를 확인합니다. *password*

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 계정이 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 계정을 구성한 후에는 이 계정을 만료되지 않도록 재구성할 수 없습니다. 단, 사용 가능한 최신 만료일의 계정을 구성할 수 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** 트랜잭션을 커밋합니다.
Firepower-chassis security/local-user # **commit-buffer**

다음의 예는 kikipopo라는 이름의 사용자 계정을 생성하고 이 사용자 계정을 활성화하며 비밀번호를 foo12345로 설정하며 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* # commit-buffer
Firepower-chassis /security/local-user #
```

다음의 예는 lincey라는 이름의 사용자 계정을 생성하고 이 사용자 계정을 활성화하며 비밀번호 없는 액세스에 사용되는 OpenSSH 키를 설정하며 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* # commit-buffer
Firepower-chassis /security/local-user #
```

다음의 예는 jforlenz라는 이름의 사용자 계정을 생성하고 이 사용자 계정을 활성화하며 비밀번호 없는 액세스에 사용되는 보안 SSH 키를 설정하며 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 계정 삭제

절차

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

단계 **2** 로컬 사용자 계정을 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 계정을 삭제하고 트랜잭션을 커밋합니다.

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

로컬 사용자 계정 활성화 또는 비활성화

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

절차

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

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

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

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

다음 예에서는 계정 관리라고 하는 로컬 사용자 계정을 활성화합니다.

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

로컬에서 인증된 사용자용 비밀번호 기록 지우기

절차

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

단계 **2** 지정된 사용자 계정에 대해 로컬 사용자 보안 모드를 입력합니다.
Firepower-chassis /security # **scope local-user** *user-name*

단계 **3** 지정된 사용자 계정에 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 기록 수를 구성하고 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

FXOS 섀시는 다음의 2가지 기본 이미지 유형을 사용합니다.

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

- 플랫폼 번들 - Firepower 플랫폼 번들은 Firepower 수퍼바이저 및 Firepower 보안 모듈/엔진에서 작동하는 여러 개별 이미지의 집합입니다. 플랫폼 번들은 Firepower eXtensible 운영 체제 소프트웨어 패키지입니다.
- 애플리케이션 - 애플리케이션 이미지는 보안 모듈/엔진(FXOS 섀시)에 구축하려는 소프트웨어 이미지입니다. 애플리케이션 이미지는 CSP(Cisco Secure Package) 파일로 전송되고 논리적 디바이스 생성의 일부로 또는 이후의 논리적 디바이스 생성을 준비하기 위해 보안 모듈/엔진에 구축될 때까지 수퍼바이저에 저장됩니다. 릴리스 1.1.1의 경우, 사용 가능한 유일한 애플리케이션 이미지는 ASA용입니다.Firepower 수퍼바이저에 저장된 동일한 애플리케이션 이미지 유형의 여러 가지 다른 버전이 있을 수 있습니다.

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

Cisco.com에서 이미지 다운로드

시작하기 전에

Cisco.com 계정이 있어야 합니다.

절차

단계 **1** 웹 브라우저를 사용하여 http://www.cisco.com/go/firepower9300-software 또는 http://www.cisco.com/go/firepower9300-software으로 이동합니다.
FXOS 섀시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.

단계 **2** 적절한 소프트웨어 이미지를 찾은 다음 로컬 컴퓨터에 다운로드합니다.

Firepower eXtensible 운영 체제 소프트웨어 이미지를 FXOS 섀시에 다운로드

FTP, SCP, SFTP 또는 TFTP를 사용하여 FXOS 소프트웨어 이미지를 FXOS 섀시에 복사할 수 있습니다.

시작하기 전에

컨피그레이션 파일을 가져오는 데 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 입력합니다.
Firepower-chassis # **scopefirmware**

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

단계 **3** 다운로드 프로세스를 모니터링하려면 다음을 수행합니다.
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)
```

Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드

시작하기 전에

Cisco.com에서 플랫폼 번들 소프트웨어 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 38 페이지 참고)한 다음 해당 이미지를 다운로드합니다(대상 위치: FXOS 섀시, FXOS 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 40 페이지 참고).

절차

단계 **1** FXOS CLI에 연결합니다(FXOS CLI 액세스, 12 페이지 참고).

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

단계 **3** 자동 설치 모드를 입력합니다.
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** **yes**를 입력하여 설치를 계속할 것인지 확인하거나 **no**를 입력하여 설치를 취소합니다.

Firepower eXtensible 운영 체제에서는 번들의 압축을 풀고 구성 요소를 업그레이드 및 다시 로드합니다.

단계 **6** 업그레이드 프로세스를 모니터링하려면 다음을 수행합니다.

a) **scopefirmware**를 입력합니다.
b) **scopeauto-install**을 입력합니다.
c) **showfsmstatusexpand**를 입력합니다.

FXOS 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드

FTP, SCP, SFTP 또는 TFTP를 사용하여 논리적 디바이스 소프트웨어 이미지를 FXOS 섀시에 복사할 수 있습니다.

시작하기 전에

컨피그레이션 파일을 가져오는 데 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 입력합니다.
Firepower-chassis # **scopessa**

단계 **2** 애플리케이션 소프트웨어 모드를 입력합니다.
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** 다운로드 프로세스를 모니터링하려면 다음을 수행합니다.
Firepower-chassis /ssa/app-software # **show download-task**

단계 **5** 다운로드한 애플리케이션을 보려면 다음을 수행합니다.
Firepower-chassis /ssa/app-software # **up**

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

단계 **6** 특정 애플리케이션에 대한 세부사항을 보려면 다음을 수행합니다.
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 디바이스용 이미지 버전 업데이트

시작하기 전에

Cisco.com에서(Cisco.com에서 이미지 다운로드, 38 페이지 참고) 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음 해당 이미지를 다운로드합니다(대상 위치: FXOS 섀시, FXOS 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 40 페이지 참고).

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

절차

단계 **1** 보안 서비스 모드를 입력합니다.
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 업데이트하려는 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 ASA용 소프트웨어 이미지를 업데이트합니다. 참고로, 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 #
```

7 장

플랫폼 설정

관리 IP 주소 변경

시작하기 전에

FXOS 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 연결을 다시 설정해야 합니다.

절차

단계 1 FXOS CLI에 연결합니다(FXOS CLI 액세스, 12 페이지 참고).

단계 2 IPv4 관리 IP 주소를 구성하려면 다음을 수행합니다.

a) Fabric-interconnect a에 대한 범위를 설정합니다.

Firepower-chassis# **scope fabric-interconnect a**

b) 현재 관리 IP 주소를 보려면 다음 명령을 입력합니다.
Firepower-chassis /fabric-interconnect # **show**

c) 새로운 관리 IP 주소 및 게이트웨이를 구성하려면 다음 명령을 입력합니다.
Firepower-chassis /fabric-interconnect # **set out-of-band ip** *ip_address* **netmask** *network_mask* **gw** *gateway_ip_address*

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

단계 **3** IPv6 관리 IP 주소를 구성합니다.

a) Fabric-interconnect a에 대한 범위를 설정합니다.
Firepower-chassis# **scope fabric-interconnect a**

b) 관리 IPv6 컨피그레이션에 대한 범위를 설정합니다.
Firepower-chassis /fabric-interconnect # **scope ipv6-config**

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

d) 새로운 관리 IP 주소 및 게이트웨이를 구성하려면 다음 명령을 입력합니다.
Firepower-chassis /fabric-interconnect/ipv6-config # **set out-of-band ipv6** *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 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 ipv6 2001::8999
ipv6-prefix 64 ipv6-gw 2001::1
Firepower-chassis /fabric-interconnect/ipv6-config* # commit-buffer
Firepower-chassis /fabric-interconnect/ipv6-config #
```

날짜 및 시간 설정

아래에 설명된 CLI 명령을 사용하여 날짜 및 시간을 수동으로 설정하거나 NTP 서버를 구성할 수 있습니다.

참고 NTP 설정은 섀시에 설치된 Firepower 섀시와 애플리케이션 간에 동기화되지 않습니다. 올바른 작동을 위해 Firepower 섀시 및 섀시에서 실행되는 애플리케이션에 동일한 NTP 설정을 구성해야 합니다.

표준 시간대 설정

절차

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

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

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

이때 사용자의 대륙, 국가 및 표준 시간대 영역에 해당하는 숫자를 입력하라는 메시지가 표시됩니다. 각 메시지에 적절한 정보를 입력합니다.

위치 정보 지정을 완료한 경우 올바른 표준 시간대 정보를 설정 중인지 확인하라는 메시지가 표시됩니다. 확인하려면 1(예)을 입력하거나 작업을 취소하려면 2(아니오)를 입력합니다.

단계 **4** 구성된 표준 시간대를 확인하려면 다음을 수행합니다.
Firepower-chassis /system/services # **top**
Firepower-chassis# **show timezone**

다음의 예에서는 표준 시간대를 태평양 표준 시간대 영역으로 구성하고 트랜잭션을 커밋하며 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#
```

NTP 서버 추가

절차

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

단계 **2** 시스템 서비스 모드를 입력합니다.
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**

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 입력합니다.
Firepower-chassis# **scopesystem**

단계 **2** 시스템 서비스 모드를 입력합니다.
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 직접 설정

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

참고 시스템 클록을 NTP 서버와 현재 동기화하는 경우, 날짜 및 시간을 수동으로 설정할 수 없습니다.

절차

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

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

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

월의 경우, 월의 처음 세자리를 사용합니다. 시간은 24시간 형식을 사용하여 입력해야 하며 이때 오후 7시는 19로 입력됩니다.

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 방법을 설명합니다. SSH는 기본적으로 활성화되어 있습니다.

절차

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

단계 **2** 시스템 서비스 모드를 입력합니다.
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**

다음의 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 통해서만 사용할 수 있습니다.

절차

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

단계 **2** 시스템 서비스 모드를 입력합니다.
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**

다음의 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 방법에 대해 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

SNMP 소개

SNMP(Simple Network Management Protocol)는 SNMP 관리자 및 에이전트 간 통신에 메시지 형식을 제공하는 애플리케이션 레이어 프로토콜입니다. SNMP는 네트워크에 있는 디바이스의 모니터링 및 관리에 사용되는 표준화된 프레임워크 및 공통 언어를 제공합니다.

SNMP 프레임워크는 다음 3가지 부분으로 구성됩니다.

- SNMP 관리자 - SNMP를 사용하여 네트워크 디바이스 활동을 제어하고 모니터링하는 데 사용되는 시스템입니다.
- SNMP 에이전트 - Firepower 섀시용 데이터를 유지 관리하고 필요 시 데이터를 SNMP 관리자에 보고하는 Firepower 섀시에 포함된 소프트웨어 구성 요소입니다. Firepower 섀시는 에이전트 및 MIB 집합을 포함합니다. SNMP 에이전트를 활성화하고 관리자 및 에이전트 간의 관계를 생성하려면 Firepower Chassis Manager 또는 FXOS CLI에서 SNMP를 활성화하고 구성하십시오.

- MIB(관리 정보 데이터베이스) - SNMP 에이전트에 있는 관리 객체의 집합입니다.

Firepower 섀시는 SNMPv1, SNMPv2c 및 SNMPv3를 지원합니다. SNMPv1 및 SNMPv2c 모두 보안 커뮤니티 기반 양식을 사용합니다. SNMP는 다음에 정의되어 있습니다.

- 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 주요 기능은 SNMP 에이전트에서 알림을 생성하는 기능입니다. 이러한 알림에는 SNMP 관리자로부터 전송된 요청이 필요하지 않습니다. 알림은 잘못된 사용자 인증, 재시작, 연결 종료, 인접 라우터에 대한 연결 손실 또는 기타 중요한 이벤트를 나타낼 수 있습니다.

Firepower 섀시는 트랩 또는 알림 중 하나로 SNMP 알림을 생성합니다. 트랩은 SNMP 관리자가 트랩을 수신할 때 승인을 전송하지 않기 때문에 알림보다 신뢰도가 낮으며 Firepower 섀시는 트랩 수신 여부를 결정할 수 없습니다. 알림 요청을 수신하는 SNMP 관리자는 SNMP 응답 PDU(Protocol Data Unit)가 있는 메시지를 승인합니다. Firepower 섀시가 PDU를 수신하지 못하는 경우 알림 요청을 다시 전송할 수 있습니다.

SNMP 보안 수준 및 권한

SNMPv1, SNMPv2c 및 SNMPv3는 각각 다른 보안 모델을 나타냅니다. 보안 모델은 선택한 보안 수준에 결합되어 SNMP 메시지를 처리할 때 적용된 보안 메커니즘을 결정합니다.

보안 수준은 SNMP 트랩에 연결된 메시지를 표시하는 데 필요한 권한을 결정합니다. 권한 수준은 메시지가 공개되는 것으로부터 보호되어야 하는지 또는 인증되어야 하는지 여부를 결정합니다. 지원되는 보안 수준은 어떤 보안 모델이 구현되는지에 따라 다릅니다. SNMP 보안 수준은 다음 권한 중 하나 이상을 지원합니다.

- noAuthNoPriv - 인증 또는 암호화 없음
- authNoPriv - 인증하지만 암호화 없음

• authPriv - 인증 및 암호화

SNMPv3는 보안 모델 및 보안 수준 모두를 위해 제공됩니다. 보안 모델은 사용자 및 사용자 역할에 대해 설정된 인증 전략입니다. 보안 수준은 보안 모델에서 허용된 보안 수준입니다. 보안 모델 및 보안 수준의 조합은 SNMP 패킷을 처리할 때 적용할 보안 메커니즘을 결정합니다.

SNMP 보안 모델과 수준의 지원되는 조합

다음 표에서는 보안 모델과 수준의 조합이 무엇을 의미하는지 확인할 수 있습니다.

표 *4: SNMP* 보안 모델과 수준

모델	수준	인증	암호화	결과
v1	noAuthNoPriv	커뮤니티 문자열	아니요	인증에 커뮤니티 문자열 일치를 사용합니다.
v2c	noAuthNoPriv	커뮤니티 문자열	아니요	인증에 커뮤니티 문자열 일치를 사용합니다.
v3	noAuthNoPriv	Username	아니요	인증에 사용자 이름 일치를 사용합니다.
v3	authNoPriv	HMAC-SHA	아니요	HMAC SHA(보안 해시 알고리즘) 기반 인증을 제공합니다.
v3	authPriv	HMAC-SHA	DES	HMAC-SHA 알고리즘 기반 인증을 제공합니다. CBC(Cipher Block Chaining) DES(DES-56) 표준 기반 인증 외에 DES(데이터 암호화 표준) 56비트 암호화를 제공합니다.

SNMPv3 보안 기능

SNMPv3는 네트워크에서 인증 및 암호화 프레임을 조합하여 디바이스에 대한 보안 액세스를 제공합니다. SNMPv3는 구성된 사용자가 수행하는 관리 작업만 승인하고 SNMP 메시지를 암호화합니다. SNMPv3 USM(사용자 기반 보안 모델)은 SNMP 메시지 수준 보안을 참고하며 다음 서비스를 제공합니다.

- 메시지 무결성 - 메시지가 무단으로 변경 또는 손상되지 않았으며 데이터 시퀀스가 악의 없이 수행될 수 있었던 것보다 더 적게 변경되었음을 보장합니다.
- 메시지 원본 인증 - 수신한 데이터의 출처를 대신하는 사용자의 요청된 ID가 확인되었음을 보장합니다.
- 메시지 기밀성 및 암호화 - 권한 없는 개인, 엔터티 또는 프로세스에서 정보를 사용할 수 없거나 정보를 공개하지 않았음을 보장합니다.

SNMP 지원

Firepower 섀시는 SNMP에 다음 지원을 제공합니다.

MIB 지원

Firepower 섀시는 MIB에 대해 읽기 전용 액세스를 지원합니다.

SNMPv3 사용자의 인증 프로토콜

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

SNMPv3 사용자의 AES 프라이버시 프로토콜

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

프라이버시 비밀번호 또는 priv 옵션은 SNMP 보안 암호화를 위해 DES 선택사항 또는 128비트 AES 암호화를 제공합니다. AES-128 컨피그레이션을 활성화하고 SNMPv3 사용자에 대한 프라이버시 비밀번호를 포함하는 경우, Firepower 섀시는 프라이버시 비밀번호를 사용하여 128비트 AES 키를 생성합니다. AES 프라이버시 비밀번호는 8자 이상이어야 합니다. 암호가 일반 텍스트에서 지정된 경우, 최대 64자를 지정할 수 있습니다.

SNMP 활성화 및 SNMP 속성 구성

절차

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

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

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

set snmp community 명령을 입력하면 SNMP 커뮤니티를 시작할지 묻는 메시지가 표시됩니다.

단계 **4** SNMP 커뮤니티를 지정합니다. 커뮤니티 이름을 비밀번호로 사용합니다. 커뮤니티 이름에는 최대 32자의 영숫자 문자열을 사용할 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 전화 번호로, 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 이름에는 최대 512자의 영숫자 문자열을 사용할 수 있습니다.
Firepower-chassis /monitoring # **set snmp syslocation***system-location-name*

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

다음의 예에서는 SNMP를 활성화하고 SNMP 커뮤니티 이름인 SnmpCommSystem2를 구성하고 시스템 담당자 이름인 contactperson을 구성하고 연락처 위치 이름인 systemlocation을 구성하고 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 입력합니다.
Firepower-chassis# **scope monitoring**

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

단계 **3** 지정된 호스트 이름, IPv4 주소 또는 IPv6 주소가 있는 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**}

결과:

- v2c 또는 v3를 버전으로 선택한 경우 **traps**
- v2c를 버전으로 선택한 경우 **informs**

참고 알림 공지는 v2c를 버전으로 선택한 경우에만 전송할 수 있습니다.

단계 **8** (선택 사항) v3를 버전으로 선택한 경우 트랩과 연관된 권한을 지정합니다.
Firepower-chassis /monitoring/snmp-trap # **set v3privilege** {**auth** | **noauth** | **priv**}

결과:

- **auth** - 인증하지만 암호화 없음
- **noauth** - 인증 또는 암호화 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 해당 트랩이 포트 2에서 SnmpCommSystem2 커뮤니티를 사용할 것임을 지정하고 버전을 v3로 설정하고 알림 유형을 트랩으로 설정하고 v3 권한을 priv로 설정하며 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 해당 트랩이 포트 2에서 SnmpCommSystem3 커뮤니티를 사용할 것임을 지정하고 버전을 v3로 설정하고 알림 유형을 트랩으로 설정하고 v3 권한을 priv로 설정하며 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 입력합니다.
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 192.168.100.112에서 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 입력합니다.
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 입력한 후 비밀번호를 입력하라는 메시지가 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 프라이버시 비밀번호를 지정합니다.
Firepower-chassis /monitoring/snmp-user # **set priv-password**

set priv-password 명령을 입력한 후 프라이버시 비밀번호를 입력하고 확인하라는 프롬프트가 표시됩니다.

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

다음의 예에서는 SNMP를 활성화하고 snmp-user14라는 이름의 SNMPv3 사용자를 생성하고 AES-128 암호화를 활성화하며 비밀번호 및 프라이버시 비밀번호를 설정하며 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 입력합니다.
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 포트 변경

HTTPS 서비스는 기본적으로 포트 443에서 활성화되어 있습니다. HTTPS를 비활성화할 수는 없지만, HTTPS 연결에 사용할 포트를 변경할 수 있습니다.

절차

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

단계 **2** 시스템 서비스 모드를 입력합니다.
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 닫힙니다. 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.

다음의 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 #
```

AAA 구성

이 섹션에서는 인증, 권한 부여 및 계정 관리에 대해 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

AAA 정보

AAA는 컴퓨터 리소스에 대한 액세스 제어를 위한 서비스 집합으로, 정책을 구현하고 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 저장된 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 FXOS 섀시를 구성할 수 있습니다.

- HTTPS
- SSH
- 시리얼 콘솔

승인

승인은 정책을 구현하는 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.

회계

어카운팅은 사용자가 액세스 중 소비하는 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 승인 제어, 청구, 경향 분석, 리소스 활용도 및 용량 계획 활동에 사용됩니다.

인증, 권한 부여 및 계정 관리 간 상호 작용

인증을 단독으로 사용하거나 권한 부여 및 어카운팅과 함께 사용할 수 있습니다. 인증은 항상 사용자를 먼저 확인해야 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. 인증은 사용자가 액세스할 수 있는 리소스와 서비스를 결정하는 정책을 구현합니다. 어카운팅은 청구 및 분석을 위해 사용되는 시간과 데이터를 추적합니다.

로컬 데이터베이스 지원

Firepower 섀시는 사용자가 사용자 프로필을 채울 수 있는 로컬 데이터베이스를 유지합니다. 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 계정을 생성하여 Firepower eXtensible 운영 체제와 바인딩합니다. 이 어카운트에는 만료되지 않는 비밀번호가 제공되어야 합니다.

절차

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

단계 **2** 보안 LDAP 모드를 입력합니다.
Firepower-chassis /security # **scope ldap**

단계 **3** 지정된 특성을 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 특성을 CiscoAvPair로, 기본 고유 이름을 "DC=cisco-firepower-aaa3,DC=qalab,DC=com"으로, 필터를 sAMAccountName=$userid로, 시간 제한 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 계정을 생성하여 Firepower eXtensible 운영 체제와 바인딩합니다. 이 어카운트에는 만료되지 않는 비밀번호가 제공되어야 합니다.

절차

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

단계 **2** 보안 LDAP 모드를 입력합니다.
Firepower-chassis /security # **scope ldap**

단계 **3** LDAP 서버 인스턴스를 생성하고 보안 LDAP 서버 모드를 입력합니다.
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(공통 이름)과 정확하게 일치해야 합니다. IP 주소가 지정되지 않은 한, DNS 서버를 구성해야 합니다.

단계 **4** (선택 사항) 사용자 역할 및 로케일에 대한 값을 저장하는 LDAP 특성을 설정합니다.
Firepower-chassis /security/ldap/server # **set attribute***attr-name*

이 속성은 항상 이름 값 쌍입니다. 시스템은 이 특성 이름과 일치하는 값에 대해 사용자 레코드를 쿼리합니다.

이 값은 기본 특성이 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 DN의 길이는 최대 255자에서 CN=username 길이를 뺀 문자 수로 설정할 수 있습니다. 이때 사용자 이름은 LDAP 인증을 사용하여 Firepower Chassis Manager 또는 FXOS CLI에 액세스를 시도하는 원격 사용자를 식별합니다.

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 권한을 지닌 LDAP 데이터베이스 계정의 DN(고유 이름)을 설정합니다.
Firepower-chassis /security/ldap/server # **set binddn***binddn-name*

지원되는 최대 문자열 길이는 255개의 ASCII 문자입니다.

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 경우에 필요합니다.

단계 **8** 바인드 DN에 지정된 LDAP 데이터베이스 계정의 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 **Enter**를 누르고(**set password** 명령 입력 후) 메시지에 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 때 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 암호화되지 않은 텍스트로 전송됩니다.

LDAP은 STARTTLS를 사용합니다. 이는 포트 389를 사용하여 암호화된 통신을 허용합니다.

단계 **12** 시간이 초과되기 전에 시스템이 LDAP 데이터베이스에 연결을 시도하는 데 사용해야 하는 시간(초)을 지정합니다.
Firepower-chassis /security/ldap/server # **set timeout***timeout-num*

1~60(초)의 정수를 입력하거나 0(영)을 입력하여 LDAP 제공자에 지정된 글로벌 시간 제한 값을 사용합니다. 기본값은 30초입니다.

단계 **13** LDAP 제공자 또는 서버의 세부 정보를 제공하는 벤더를 지정합니다.
Firepower-chassis /security/ldap/server # **set vendor***{ms-ad | openldap}*

옵션은 다음과 같습니다.

- **ms-ad** - LDAP 제공자가 Microsoft Active Directory임
- **openldap** - LDAP 제공자가 Microsoft Active Directory가 아님

단계 **14** 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 특성을 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 특성을 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 입력합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 LDAP 모드를 입력합니다.
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 입력합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 RADIUS 모드를 입력합니다.
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 제한 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 체제에서는 최대 16개의 RADIUS 제공자를 지원합니다.

절차

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

단계 **2** 보안 RADIUS 모드를 입력합니다.
Firepower-chassis /security # **scope radius**

단계 **3** RADIUS 서버 인스턴스를 생성하고 보안 RADIUS 서버 모드를 입력합니다.
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 **Enter**를 누르고(**set key** 명령 입력 후) 프롬프트에서 키 값을 입력합니다.

단계 **6** (선택 사항) 이 서버를 순서대로 시도할 시기를 지정합니다.
Firepower-chassis /security/radius/server # **set order** *order-num*

단계 **7** (선택 사항) 서버가 다운되었다고 인지할 때까지 RADIUS 서버와의 통신을 재시도한 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 제공자를 위해 2가지 계수 인증을 선택한 경우 더 높은 **Timeout(**시간 제한**)** 값을 구성할 것을 권장합니다.

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 2가지 계수 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 입력합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 RADIUS 모드를 입력합니다.
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 입력합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 TACACS+ 모드를 입력합니다.
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**

다음의 예에서는 TACACS+ 시간 제한 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 체제에서는 최대 16개의 TACACS+ 제공자를 지원합니다.

절차

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

단계 **2** 보안 TACACS+ 모드를 입력합니다.
Firepower-chassis /security # **scope tacacs**

단계 **3** TACACS+ 서버 인스턴스를 생성하고 보안 TACACS+ 서버 모드를 입력합니다.
Firepower-chassis /security/tacacs # **create server** *server-name*

단계 **4** TACACS+ 서버 키를 지정합니다.
Firepower-chassis /security/tacacs/server # **set key**

키 값을 설정하려면 **Enter**를 누르고(**set key** 명령 입력 후) 프롬프트에서 키 값을 입력합니다.

단계 **5** (선택 사항) 이 서버를 순서대로 시도할 시기를 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+ 제공자를 위해 2가지 계수 인증을 선택한 경우 더 높은 시간 제한 값을 구성할 것을 권장합니다.

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 키를 tacacskey321로 설정하고 순서를 4로 설정하고 인증 포트를 5859로 설정하며 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 입력합니다.
Firepower-chassis# **scope security**

단계 **2** 보안 TACACS+ 모드를 입력합니다.
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 경고를 종합하는 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 입력합니다.

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 수준을 선택합니다. syslog가 활성화된 경우 시스템은 콘솔에 해당 수준 이상의 메시지를 표시합니다. 수준 옵션은 긴급도가 감소하는 순서로 나열됩니다. 기본 수준은 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 수준을 선택합니다. 모니터 상태가 활성화된 경우, 시스템에 해당 수준 이상이 표시됩니다. 수준 옵션은 긴급도가 감소하는 순서로 나열됩니다. 기본 수준은 Critical(위험)입니다.
Firepower-chassis /monitoring # **set syslog monitor level** {**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

참고 Critical(위험) 미만 수준의 메시지는 **terminal monitor** 명령을 입력한 경우에만 터미널 모니터에 표시됩니다.

단계 **6** syslog 파일에 대한 syslog 정보 작성을 활성화하거나 비활성화합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog file**

단계 **7** 메시지를 기록할 파일 이름을 지정합니다. 파일 이름에는 최대 16자를 사용할 수 있습니다.
Firepower-chassis /monitoring # **set syslog file name***filename*

단계 **8** (선택 사항) 사용자가 파일에 저장하려는 가장 낮은 메시지 수준을 선택합니다. 파일 상태가 활성화된 경우, 시스템은 syslog 파일에 해당 수준 이상의 메시지를 저장합니다. 수준 옵션은 긴급도가 감소하는 순서로 나열됩니다. 기본 수준은 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 서버로의 syslog 메시지 전송을 활성화하거나 비활성화합니다.
Firepower-chassis /monitoring # {**enable** | **disable**} **syslog remote-destination** {**server-1** | **server-2** | **server-3**}

b) (선택 사항) 사용자가 외부 로그에 저장하려는 가장 낮은 메시지 수준을 선택합니다. 원격 대상이 활성화된 경우, 시스템은 외부 서버에 해당 수준 이상의 메시지를 전송합니다. 수준 옵션은 긴급도가 감소하는 순서로 나열됩니다. 기본 수준은 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 수준을 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. 사용자가 활성화하거나 비활성화하려는 로컬 소스 각각에 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 로깅을 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 IP 주소에 대한 호스트 이름을 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. IPv4 또는 IPv6 주소 중 하나로 서버의 IP 주소를 사용해야 합니다. 최대 4개까지 DNS 서버를 구성할 수 있습니다.

참고 여러 DNS 서버를 구성할 때 시스템은 임의 순서로만 서버를 검색합니다. 로컬 관리 명령이 DNS 서버 조회를 필요로 하는 경우, 임의 순서로 3개의 DNS 서버만 검색할 수 있습니다.

절차

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

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

단계 **3** DNS 서버를 생성하거나 삭제하려면 다음과 같이 적절한 명령을 입력합니다.

- 지정된 IPv4 또는 IPv6 주소가 있는 DNS 서버를 사용하도록 시스템을 구성하려면 다음을 수행합니다.

  Firepower-chassis /system/services # **createdns***{ip-addr* | *ip6-addr}*

- 지정된 IPv4 또는 IPv6 주소가 있는 DNS 서버를 삭제하려면 다음을 수행합니다.

  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 #
```

8 장

인터페이스 관리

Firepower Security Appliance 인터페이스 정보

FXOS 섀시는 단일 인터페이스뿐만 아니라 EtherChannel(port-channel) 인터페이스를 지원합니다. EtherChannel 인터페이스는 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

각 인터페이스는 다음 유형 중 하나일 수 있습니다.

- 데이터(기본값) -- 데이터 인터페이스는 논리적 디바이스 간에 공유할 수 없습니다.
- 관리 -- 관리 인터페이스는 논리적 디바이스 간에 공유할 수 있습니다. 논리적 디바이스당 1개의 관리 인터페이스만 할당할 수 있습니다.
- 클러스터 -- 클러스터된 논리적 디바이스에 사용되는 특수 인터페이스 유형입니다. 이 유형은 유닛 클러스터 간 통신을 위한 클러스터 제어 링크에 자동으로 할당됩니다. 기본적으로, 클러스터 제어 링크는 Port-channel 48에 자동으로 생성됩니다.

인터페이스 속성 편집

절차

단계 **1** 인터페이스 모드를 입력합니다.
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 port-channel의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. 포트 채널의 멤버인 인터페이스에서 enter interface 명령을 사용하는 경우 객체가 존재하지 않는다는 오류 메시지가 표시됩니다. port-channel에 인터페이스를 추가하기 전에 enter interface 명령을 사용하여 인터페이스를 편집해야 합니다.

단계 **3** (선택 사항) 인터페이스 유형을 설정합니다.
setport-type {**data** | **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.
setspeed {**10gbps** | **1gbps**}

예제**:**

```
Firepower /eth-uplink/fabric/interface* # set speed 1gbps
```

단계 **5** 컨피그레이션을 커밋합니다.
commit-buffer

포트 채널 생성

EtherChannel(port-channel라고도 함)은 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

시작하기 전에

FXOS 섀시는 활성 LACP(링크 어그리게이션 제어 프로토콜) 모드에서 EtherChannel만 지원합니다. Cisco는 최고의 호환성을 위해 스위치 포트를 활성 모드로 연결하는 설정을 권장합니다.

절차

단계 **1** 인터페이스 모드를 입력합니다.
scopeeth-uplink

scope fabric a

단계 **2** port-channel을 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 port-channel을 기본값 대신 클러스터 제어 링크로 사용하지 않는 경우 **cluster** 키워드를 선택하지 마십시오.

단계 **5** (선택 사항) port-channel의 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

분할 케이블 구성

다음 절차에서는 FXOS 섀시에서 사용할 분할 케이블을 구성하는 방법을 보여줍니다. 분할 케이블을 사용하여 1개의 40Gbps 포트 대신 4개의 10Gbps 포트를 제공할 수 있습니다.

절차

단계 **1** 새 분할 케이블을 생성하려면 다음 명령을 사용합니다.

a) 케이블 모드를 입력합니다.
scopecabling
scope fabric a

b) 분할 케이블을 생성합니다.
createbreakout*network_module_slotport*

예제**:**

```
Firepower /cabling/fabric/ # create breakout 2 1
```

c) 컨피그레이션을 커밋합니다.
commit-buffer

자동 재부팅이 수행됩니다. 하나 이상의 분할 케이블을 생성하는 경우 commit-buffer 명령을 실행하기 전에 분할 케이블을 모두 생성해야 합니다.

단계 **2** 분할 포트를 활성화하고 구성하려면 다음 명령을 사용합니다.

a) 인터페이스 모드를 입력합니다.
scopeeth-uplink

scopefabrica

scopeaggr-interface*network_module_slotport*

b) **set** 명령을 사용하여 인터페이스 속도 및 포트 유형을 구성합니다.
enable 또는 **disable** 명령을 사용하여 인터페이스의 관리 상태를 설정합니다.

c) 컨피그레이션을 커밋합니다.
commit-buffer

9 장

논리적 디바이스

논리적 디바이스 정보

논리적 디바이스를 생성할 때 FXOS 섀시 수퍼바이저는 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 지정된 보안 모듈/엔진에 적용하거나 내장 섀시 클러스터의 경우, Firepower 섀시에 설치된 모든 보안 모듈에 적용하여 논리적 디바이스를 구축합니다.

다음 2가지 유형의 논리적 디바이스 중 하나를 생성할 수 있습니다.

참고 여러 보안 모듈을 지원하는 FXOS 섀시에서 한 가지 유형의 논리적 디바이스(독립형 또는 클러스터)만 생성할 수 있습니다. 즉, 3개의 보안 모듈이 설치된 경우, 하나의 보안 모듈에서 독립형 논리적 디바이스를 생성한 다음에 나머지 2개의 논리적 디바이스를 사용하는 클러스터를 생성할 수 없습니다.

- 독립형 - Firepower 섀시에 설치된 각각의 보안 모듈/엔진용으로 독립형 논리적 디바이스를 생성할 수 있습니다.
- 클러스터 - 클러스터링을 통해 여러 보안 모듈을 함께 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 9300 등의 여러 모듈 디바이스는 내장 섀시 클러스터링을 지원합니다.

독립형 ASA 논리적 디바이스 생성

각각의 보안 모듈/엔진용(FXOS 섀시에 설치됨)으로 독립형 논리적 디바이스를 생성할 수 있습니다. Firepower 9300과 같은 여러 모듈 디바이스에서는 클러스터가 구성되어 있는 경우 독립형 논리적 디바이스를 생성할 수 없습니다. 독립형 디바이스를 구성하려면 먼저 클러스터를 삭제해야 합니다.

시작하기 전에

- 논리적 디바이스에 사용할 보안 모듈/엔진에 논리적 디바이스가 이미 구성되어 있는 경우, 기존의 논리적 디바이스를 먼저 삭제해야 합니다(논리적 디바이스 삭제 참고).
- Cisco.com에서(Cisco.com에서 이미지 다운로드, 38 페이지 참고) 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음 해당 이미지를 다운로드합니다(대상 위치: FXOS 섀시, FXOS 섀시에 논리적 디바이스 소프트웨어 이미지 다운로드, 40 페이지 참고).
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다.

절차

단계 **1** 보안 서비스 모드를 입력합니다.
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***nameinterface_name***asa**
Firepower-chassis /ssa/logical-device/external-port-link* # **exit**

단계 **5** 관리 부트스트랩 정보를 구성합니다.

a) 부트스트랩 객체를 생성합니다.
Firepower /ssa/logical-device* # **createmgmt-bootstrapasa**

b) 비밀번호 사용을 생성합니다.
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
```

클러스터 구축

클러스터링을 통해 여러 보안 모듈을 함께 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 9300 등의 여러 모듈 디바이스는 내장 섀시 클러스터링을 지원합니다.

참고 Firepower 9300은 여러 섀시(섀시 간) 전체에서 클러스터를 지원하지 않으며 내장 섀시 클러스터링만 지원합니다.

FXOS 섀시의 클러스터링 정보

클러스터는 하나의 유닛으로 작동하는 여러 개의 디바이스로 구성됩니다. FXOS 섀시에서 클러스터를 구축하는 경우 다음 작업을 수행합니다.

- 유닛 간 통신을 위한 클러스터 제어 링크(port-channel 48)를 생성합니다. 내장 섀시 클러스터링의 경우, 이 링크는 클러스터 통신에 Firepower 9300 백플레인을 활용합니다.
- 클러스터 부트스트랩 컨피그레이션을 애플리케이션 내부에 생성합니다.

  클러스터를 구축할 때, FXOS 섀시 수퍼바이저는 클러스터 이름, 클러스터 제어 링크 인터페이스 및 기타 클러스터 설정을 포함하는 각 유닛에 최소한의 부트스트랩 컨피그레이션을 적용합니다. 클러스터링 환경을 사용자 맞춤화하려는 경우 부트스트랩 컨피그레이션의 일부는 애플리케이션 내부에서 사용자가 구성할 수 있습니다.
- 데이터 인터페이스를 *Spanned* 인터페이스로 클러스터에 할당합니다.

  내장 섀시 클러스터링의 경우, Spanned 인터페이스가 EtherChannels(섀시 간 클러스터링의 경우와 마찬가지로)에 국한되지 않습니다. Firepower 9300 수퍼바이저는 EtherChannel 기술을 내부에서 사용하여 트래픽을 공유 인터페이스에 있는 여러 모듈에 로드 밸런싱하므로 모든 데이터 인터페이스 유형이 Spanned 모드에서 작동합니다.

  참고 개별 인터페이스는 관리 인터페이스를 제외하고 지원되지 않습니다.
- 관리 인터페이스를 클러스터의 모든 유닛에 할당합니다.

다음 섹션에서는 클러스터링 개념 및 구현에 대한 자세한 내용을 제공합니다.

기본 유닛 및 보조 유닛 역할

클러스터의 멤버 중 하나는 기본 유닛입니다. 기본 유닛은 자동으로 결정됩니다. 다른 모든 멤버는 보조 유닛입니다.

모든 컨피그레이션은 기본 유닛에서만 수행해야 하며, 이후 컨피그레이션이 보조 유닛에 복제됩니다.

클러스터 제어 링크

클러스터 제어 링크는 Port-channel 48 인터페이스를 사용하여 자동으로 생성됩니다. 내장 섀시 클러스터링의 경우, 이 인터페이스에는 멤버 인터페이스가 없습니다. 이 클러스터 유형 EtherChannel은 내장 섀시 클러스터링을 위한 클러스터 통신에 Firepower 9300 백플레인을 활용합니다.

클러스터 제어 링크 트래픽에는 제어 및 데이터 트래픽이 모두 포함됩니다.

관리 인터페이스

클러스터에 관리 유형 인터페이스를 할당할 수 있습니다. 이 인터페이스는 Spanned 인터페이스와 반대로 특수한 개별 인터페이스입니다. 관리 인터페이스를 사용하면 각 유닛에 직접 연결할 수 있습니다.

ASA의 경우, 기본 클러스터 IP 주소는 현재 기본 유닛에 항상 속해 있는 클러스터의 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 syslog와 같은 아웃바운드 관리 트래픽의 경우 기본 유닛을 비롯한 각 유닛에서 로컬 IP 주소를 사용하여 서버에 연결합니다.

클러스터링 지침

- 이중화를 위해 EtherChannels를 VSS 또는 vPC에 연결할 것을 권장합니다.
- 섀시 내에서 일부 보안 모듈을 클러스터링하여 독립형 모드에서 다른 보안 모듈을 실행할 수 없습니다. 클러스터의 모든 보안 모듈을 포함해야 합니다.

클러스터링 기본값

클러스터 제어 링크는 Port-channel 48을 사용합니다.

ASA 클러스터링 구성

FXOS 섀시 수퍼바이저에서 손쉽게 클러스터를 구축할 수 있습니다. 모든 초기 컨피그레이션은 각 유닛에 대해 자동으로 생성됩니다.

절차

단계 **1** 클러스터를 구축하기 전에 최소 1개 이상의 데이터 유형 인터페이스 또는 EtherChannel(port-channel이라고도 함)을 구성합니다. 포트 채널 생성, 76 페이지 또는 인터페이스 속성 편집, 75 페이지를 참조하십시오.
모든 인터페이스는 클러스터에 기본적으로 할당되어 있습니다. 또한 데이터 인터페이스를 구축한 후에 클러스터에 추가할 수 있습니다.

단계 **2** 관리 유형 인터페이스 또는 EtherChannel을 추가합니다. 포트 채널 생성, 76 페이지 또는 인터페이스 속성 편집, 75 페이지를 참조하십시오.

단계 **3** Port-channel 48은 클러스터 제어 링크로 예약됩니다.

단계 **4** 보안 서비스 모드를 입력합니다.
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 FXOS 섀시 수퍼바이저가 클러스터링 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 클러스터 이름이 아닙니다. 하드웨어를 아직 설치하지 않은 경우에도 3개의 보안 모듈을 모두 지정해야 합니다.

단계 **6** 클러스터 매개변수를 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 지원되는 유일한 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. 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*

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set chassis-id 1
Firepower /ssa/logical-device/cluster-bootstrap* #
```

단계 **11** 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 비밀은 1~63자로 된 ASCII 문자열입니다. 공유 비밀은 키를 생성하는 데 사용됩니다. 이 옵션은 연결 상태 업데이트 및 전달된 패킷을 비롯한 데이터 경로 트래픽에 영향을 미치지 않으며, 항상 일반 텍스트로 전송됩니다.

단계 **12** 클러스터 부트스트랩 모드 및 논리적 디바이스 모드를 종료합니다.
exit

exit

단계 **13** 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 입력합니다.
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 **14** 컨피그레이션을 커밋합니다.
commit-buffer

FXOS 섀시 수퍼바이저는 기본 보안 모듈 소프트웨어 버전을 다운로드하고 클러스터 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 각 보안 모듈에 입력하여 클러스터를 구축합니다.

단계 **15** 클러스터링 컨피그레이션을 맞춤화하려면 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

    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
```

클러스터링 기록

기능 이름	플랫폼릴리스	기능 정보
Cisco ASA를 위한 내장 섀시 클러스터링	1.1.1	Firepower 9300 섀시 내부에서 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**

애플리케이션 콘솔 또는 데코레이터에 연결

다음 절차를 수행하여 애플리케이션 콘솔 또는 데코레이터에 연결합니다.

참고 콘솔 액세스 시 문제가 발생한 경우, 다른 SSH 클라이언트를 시도하거나 SSH 클라이언트를 새 버전으로 업그레이드할 것을 권장합니다.

절차

단계 **1** 애플리케이션 콘솔 또는 데코레이터에 연결하려면 다음을 수행합니다.

a) FXOS CLI에서 보안 모듈/엔진에 연결합니다.
Firepower-chassis # **connect module** *slot_number* **console**

참고 여러 보안 모듈을 지원하지 않는 디바이스의 보안 엔진에 연결하려면 1을 *slot_number*로 사용합니다.

보안 모듈에 처음 연결할 때, FXOS 모듈 CLI에 액세스합니다.

b) 애플리케이션 또는 데코레이터에 연결하려면 을 입력합니다.
Firepower-module1>**connect asa**

FXOS CLI의 수퍼바이저 수준에서 보안 모듈/엔진에 대한 후속 연결은 보안 모듈/엔진 OS에 직접 액세스됩니다.

단계 **2** (선택 사항) FXOS 모듈 CLI에 대한 애플리케이션 콘솔은 **Ctrl-A-D**를 입력하여 종료합니다. 문제 해결을 위해 FXOS 모듈 CLI에 액세스할 수 있습니다.

단계 **3** FXOS CLI의 수퍼바이저 수준으로 돌아갑니다.

a) 보안 모듈/엔진 콘솔을 종료하려면 ~를 입력합니다. 텔넷 애플리케이션을 종료합니다.

b) 텔넷 애플리케이션을 종료하려면 다음을 입력합니다. telnet>**quit**

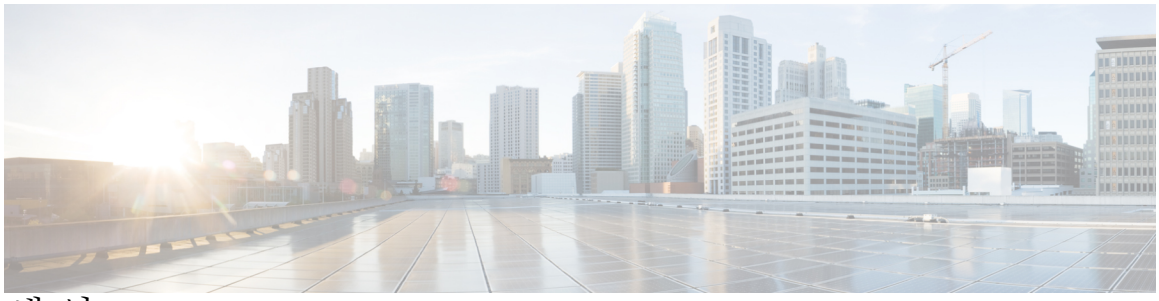
예

다음 예에서는 보안 모듈 1에 있는 ASA에 연결한 다음 FXOS CLI의 수퍼바이저 수준으로 다시 종료합니다.

```
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#
```



색 인

A

B

D

E

F

G

H

I

J

K

L

M

R

S

T

W