



## **Cisco FXOS CLI 환경 설정 가이드, 2.1(1)**

초판: 2017년 01월 23일

최종 변경: 2017년 03월 20일

### **Americas Headquarters**

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

이 설명서의 제품 관련 사양 및 정보는 예고 없이 변경될 수 있습니다. 이 설명서의 모든 설명, 정보 및 권장 사항이 정확하다고 판단되더라도 어떠한 형태의 명시적이거나 묵시적인 보증도 하지 않습니다. 모든 제품의 해당 애플리케이션에 대한 사용은 전적으로 사용자에게 책임이 있습니다.

동봉한 제품의 소프트웨어 라이선스 및 제한된 보증은 제품과 함께 제공된 정보 패키지에 설명되어 있으며 본 문서에 참조를 통해 포함됩니다. 소프트웨어 라이선스 또는 제한된 보증을 찾을 수 없는 경우 CISCO 담당자에게 문의하여 복사본을 요청하십시오.

Cisco의 TCP 헤더 압축은 UNIX 운영 체제의 UCB 공개 도메인 버전의 일부로서 UCB(University of Berkeley)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 침해 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 없으므로 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 보려면 다음 URL로 이동하십시오. <http://www.cisco.com/go/trademarks> 여기에 언급된 타사 상표는 해당 소유자의 자산입니다. "파트너"라는 용어는 사용에 있어 Cisco와 기타 회사 간의 파트너 관계를 의미하지는 않습니다. (1110R)

© 2017 Cisco Systems, Inc. All rights reserved.



## 목 차

<b>Firepower Security Appliance 소개</b>	<b>1</b>
Firepower Security Appliance 정보	1
<b>CLI(Command Line Interface) 개요</b>	<b>3</b>
관리 객체	3
명령 모드	3
객체 명령	5
명령 완성	6
명령 기록	6
보류 중인 명령 커밋, 삭제 및 보기	7
CLI를 위한 온라인 도움말	7
CLI 세션 제한	7
<b>시작하기</b>	<b>9</b>
작업 흐름	9
초기 구성	10
FXOS CLI 액세스	12
<b>라이선스 관리</b>	<b>15</b>
Smart Software Licensing 정보	15
FXOS 새시의 애플리케이션용 Smart Software Licensing	16
Smart Software Manager 및 어카운트	16
가상 어카운트별로 관리되는 라이선스 및 디바이스	16
디바이스 등록 및 토큰	17
License Authority와의 정기적인 통신	17
규정 위반 상태	17
Smart Call Home 인프라	18
Smart Software Licensing 사전 요구 사항	18
Smart Software Licensing의 기본값	18
Smart Software Licensing 구성	19

- (선택 사항) HTTP 프록시 구성 19
- (선택 사항) Call Home URL 삭제 20
- License Authority에 Firepower Security Appliance 등록 20
- FXOS 새시의 Smart License Manager Satellite 21
- FXOS 새시의 Smart License Satellite Server 구성 22
- 영구 라이선스 예약 23
  - 영구 라이선스 예약 구성 23
  - (선택 사항) FXOS 영구 라이선스 반환 24
- Smart Software Licensing 모니터링 24
- Smart Software Licensing 기록 25
- 사용자 관리 27
  - 사용자 계정 28
  - 사용자 이름 지침 29
  - 비밀번호 지침 29
  - 원격 인증에 대한 지침 30
  - 사용자 역할 33
  - 로컬 인증 사용자에게 대한 비밀번호 프로필 33
  - 기본 인증 서비스 선택 34
  - 절대 세션 시간 초과 구성 36
  - 원격 사용자에게 대한 역할 정책 구성 36
  - 로컬로 인증된 사용자의 비밀번호 보안 수준 확인 활성화 37
  - 최대 로그인 시도 횟수 설정 38
  - 변경 간격에 대해 최대 비밀번호 변경 횟수 구성 39
  - 최소 비밀번호 길이 확인 구성 39
  - 비밀번호에 대해 변경 간격 없음 구성 40
  - 비밀번호 기록 수 구성 41
  - 로컬 사용자 계정 생성 41
  - 로컬 사용자 계정 삭제 43
  - 로컬 사용자 계정 활성화 또는 비활성화 44
  - 로컬 인증 사용자에게 대한 비밀번호 기록 지우기 44
- 이미지 관리 47
  - 이미지 관리 정보 47

- Cisco.com에서 이미지 다운로드 48
- Firepower eXtensible 운영 체제 소프트웨어 이미지를 FXOS 새시에 다운로드 48
- 이미지의 무결성 확인 49
- Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드 50
- FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드 50
- 논리적 디바이스를 위한 이미지 버전 업데이트 53
- 펌웨어 업그레이드 54
- 보안 인증 컴플라이언스 57
  - 보안 인증 컴플라이언스 57
  - FIPS 모드 활성화 58
  - Common Criteria 모드 활성화 59
  - SSH 호스트 키 생성 59
  - IPSec 보안 채널 구성 60
  - 트러스트 포인트에 대한 정적 CRL 구성 65
  - 인증서 해지 목록 확인 정보 66
  - CRL 주기적 다운로드 구성 70
  - NTP 서버 인증 활성화 71
  - LDAP 키 링 인증서 설정 72
  - IP 액세스 목록 구성 73
  - 클라이언트 인증서 인증 활성화 74
- 시스템 관리 77
  - 관리 IP 주소 변경 77
  - 애플리케이션 관리 IP 변경 79
  - Pre-Login 배너 80
    - Pre-Login 배너 생성 80
    - Pre-Login 배너 수정 81
    - Pre-Login 배너 삭제 82
  - FXOS 새시 리부팅 83
  - FXOS 새시 전원 끄기 83
  - 신뢰할 수 있는 ID 인증서 설치 84
  - 소프트웨어 장애에서 복구 89
  - 손상된 파일 시스템에서 복구 93

- 플랫폼 설정 103
  - 날짜 및 시간 설정 103
    - 구성된 날짜 및 시간 보기 104
    - 표준 시간대 설정 104
    - NTP를 사용하여 날짜 및 시간 설정 106
    - NTP 서버 삭제 107
    - 날짜 및 시간 직접 설정 107
- SSH 구성 108
- 텔넷 구성 109
- SNMP 구성 110
  - SNMP 정보 110
  - SNMP 알림 111
  - SNMP 보안 레벨 및 권한 111
  - SNMP 보안 모델 및 레벨의 지원되는 조합 111
  - SNMPv3 보안 기능 112
  - SNMP 지원 112
  - SNMP 활성화 및 SNMP 속성 구성 113
  - SNMP 트랩 생성 114
  - SNMP 트랩 삭제 115
  - SNMPv3 사용자 생성 116
  - SNMPv3 사용자 삭제 117
- HTTPS 구성 117
  - 인증서, 키 링, 신뢰 지점 117
  - 키 링 생성 118
  - 기본 키 링 재생성 119
  - 키 링에 대한 인증서 요청 생성 119
    - 기본 옵션으로 키 링에 대한 인증서 요청 생성 119
    - 고급 옵션으로 키 링에 대한 인증서 요청 생성 120
  - 신뢰 지점 생성 122
  - 키 링으로 인증서 가져오기 123
- HTTPS 구성 124
  - HTTPS 포트 변경 126

- 키 링 삭제 126
- 신뢰 지점 삭제 127
- HTTPS 비활성화 127
- AAA 구성 128
  - AAA 정보 128
  - LDAP 제공자 구성 129
    - LDAP 제공자의 속성 구성 129
    - LDAP 제공자 생성 130
    - LDAP 제공자 삭제 133
  - RADIUS 제공자 구성 133
    - RADIUS 제공자의 속성 구성 133
    - RADIUS 제공자 생성 134
    - RADIUS 제공자 삭제 135
  - TACACS+ 제공자 구성 136
    - TACACS+ 제공자의 속성 구성 136
    - TACACS+ 제공자 생성 136
    - TACACS+ 제공자 삭제 137
- Syslog 구성 138
- DNS 서버 구성 140
- 인터페이스 관리 143
  - Firepower Security Appliance 인터페이스 정보 143
    - 인터페이스 유형 143
    - 하드웨어 바이패스 쌍 144
    - Jumbo Frame Support 145
  - 인터페이스 속성 편집 145
  - 포트 채널 생성 146
  - 분할 케이블 구성 147
- 논리적 디바이스 149
  - 논리적 디바이스 정보 149
  - 독립형 ASA 논리적 디바이스 생성 150
  - 독립형 위협 방어 논리적 디바이스 생성 152
  - 클러스터 구축 156

- FXOS 새시의 클러스터링 정보 156
  - 기본 유닛 및 보조 유닛 역할 157
  - 클러스터 제어 링크 157
    - 새시 간 클러스터링을 위한 클러스터 제어 링크 크기 조정 157
    - 새시 간 클러스터링을 위한 클러스터 제어 링크 이중화 158
    - 새시 간 클러스터링을 위한 클러스터 제어 링크 안정성 158
  - 관리 네트워크 158
  - 관리 인터페이스 159
  - Spanned EtherChannel 159
  - 사이트 사이트별 MAC 및 IP 주소 160
  - 클러스터링의 사전 요구 사항 160
  - 클러스터링 지침 161
  - 클러스터링 기본값 164
  - ASA 클러스터링 구성 164
  - Firepower Threat Defense 클러스터링 구성 170
  - 클러스터링 기록 178
  - 서비스 체이닝 정보 179
  - 독립형 논리적 디바이스에 Radware DefensePro 서비스 체인 구성 182
  - 내장 새시 클러스터에 Radware DefensePro 서비스 체인 구성 183
  - UDP/TCP 포트 열기 및 vDP 웹 서비스 활성화 186
  - 애플리케이션 콘솔 또는 데코레이터에 연결 187
  - 논리적 디바이스 삭제 188
  - 논리적 디바이스와 연결되지 않은 애플리케이션 인스턴스 삭제 189
- 보안 모듈/엔진 관리 191
  - FXOS 보안 모듈/보안 엔진 정보 191
  - 보안 모듈 해제/재위탁 192
  - 보안 모듈/엔진 확인 193
  - 보안 모듈/엔진 확인 재설정 194
  - 보안 모듈/엔진 확인 다시 초기화 195
  - 보안 모듈/엔진 전원 켜기/끄기 196
- 컨피그레이션 가져오기/내보내기 199
  - 컨피그레이션 가져오기/내보내기 정보 199









# Firepower Security Appliance 소개

- [Firepower Security Appliance 정보, 1 페이지](#)

## Firepower Security Appliance 정보

Cisco FXOS 새시는 네트워크 및 콘텐츠 보안 솔루션을 위한 차세대 플랫폼입니다. FXOS 새시는 Cisco ACI(Application Centric Infrastructure) 보안 솔루션에 포함되며 확장성, 제어 일관성 및 관리 간소화를 위해 구축된 민첩한 개방형 보안 플랫폼을 제공합니다.

FXOS 새시에서 제공하는 기능은 다음과 같습니다.

- 모듈형 새시 기반 보안 시스템 — 고성능의 유연한 입/출력 컨피그레이션 및 확장성을 제공합니다.
- Firepower Chassis Manager — 그래픽 사용자 인터페이스는 현재 새시 상태를 간단하게 시각적으로 표시하며 간소화된 새시 기능 컨피그레이션을 제공합니다.
- FXOS CLI — 기능 구성, 새시 상태 모니터링 및 고급 문제 해결 기능 액세스를 위해 명령어 기반 인터페이스를 제공합니다.
- FXOS REST API — 사용자는 새시를 프로그래밍 방식으로 구성 및 관리할 수 있습니다.





## CLI(Command Line Interface) 개요

- 관리 객체, 3 페이지
- 명령 모드, 3 페이지
- 객체 명령, 5 페이지
- 명령 완성, 6 페이지
- 명령 기록, 6 페이지
- 보류 중인 명령 커밋, 삭제 및 보기, 7 페이지
- CLI를 위한 온라인 도움말, 7 페이지
- CLI 세션 제한, 7 페이지

### 관리 객체

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

관리 객체에는 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 나뉩니다. **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
EXEC	모든 모드의 <b>top</b> 명령	#
어댑터	EXEC 모드의 <b>scope adapter</b> 명령	/adapter #
케이블링	EXEC 모드의 <b>scope cabling</b> 명령	/cabling #
새시	EXEC 모드의 <b>scope chassis</b> 명령	/chassis #
이더넷 서버	EXEC 모드의 <b>scope eth-server</b> 명령	/eth-server #
이더넷 업링크	EXEC 모드의 <b>scope eth-uplink</b> 명령	/eth-uplink #
패브릭 인터커넥트	EXEC 모드의 <b>scope fabric-interconnect</b> 명령	/fabric-interconnect #
펌웨어	EXEC 모드의 <b>scope firmware</b> 명령	/firmware #
호스트 이더넷 인터페이스	EXEC 모드의 <b>scope host-eth-if</b> 명령	/host-eth-if #
라이선스	EXEC 모드의 <b>scope license</b> 명령	/license #

모드 이름	액세스하는 데 사용된 명령	모드 프롬프트
모니터링	EXEC 모드의 <b>scope monitoring</b> 명령	/monitoring #
구성	EXEC 모드의 <b>scope org</b> 명령	/org #
보안	EXEC 모드의 <b>scope security</b> 명령	/security #
server	EXEC 모드의 <b>scope server</b> 명령	/server #
서비스 프로파일	EXEC 모드의 <b>scope service-profile</b> 명령	/service-profile #
ssa	EXEC 모드의 <b>scope ssa</b> 명령	/ssa #
system	EXEC 모드의 <b>scope system</b> 명령	/system #
가상 HBA	EXEC 모드의 <b>scope vhba</b> 명령	/vhba #
가상 NIC	EXEC 모드의 <b>scope vnic</b> 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 존재하는 지 여부에 따라 달라집니다.

표 2: 객체가 없는 경우의 일반적인 동작

명령	행동
<b>create object</b>	객체가 생성되고 해당하는 경우 컨피그레이션 모드가 시작됩니다.
<b>delete object</b>	오류 메시지가 생성됩니다.
<b>enter object</b>	객체가 생성되고 해당하는 경우 컨피그레이션 모드가 시작됩니다.
<b>scope object</b>	오류 메시지가 생성됩니다.

표 3: 객체가 있는 경우의 일반적인 동작

명령	행동
<b>create object</b>	오류 메시지가 생성됩니다.
<b>delete object</b>	객체가 삭제됩니다.
<b>enter object</b>	해당하는 경우 객체의 컨피그레이션 모드가 시작됩니다.
<b>scope object</b>	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 체제는 한 번에 활성화할 수 있는 CLI 세션의 수를 총 32개로 제한합니다. 이 값을 구성할 수 없습니다.





## 시작하기

---

- [작업 흐름, 9 페이지](#)
- [초기 구성, 10 페이지](#)
- [FXOS CLI 액세스, 12 페이지](#)

## 작업 흐름

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

### 절차

- 단계 1** FXOS 새시 하드웨어를 구성합니다([Cisco Firepower Security Appliance 하드웨어 설치 가이드](#) 참고).
  - 단계 2** 초기 컨피그레이션을 완료합니다([초기 구성, 10 페이지](#) 참고).
  - 단계 3** 날짜 및 시간을 설정합니다([날짜 및 시간 설정, 103 페이지](#) 참고).
  - 단계 4** DNS 서버를 구성합니다([DNS 서버 구성, 140 페이지](#) 참고).
  - 단계 5** 제품 라이선스를 등록합니다([라이선스 관리, 15 페이지](#) 참고).
  - 단계 6** 사용자를 구성합니다([사용자 관리, 27 페이지](#) 참고).
  - 단계 7** 필요 시 소프트웨어 업데이트를 수행합니다([이미지 관리, 47 페이지](#) 참고).
  - 단계 8** 추가 플랫폼 설정을 구성합니다([플랫폼 설정, 103 페이지](#) 참고).
  - 단계 9** 인터페이스를 구성합니다([인터페이스 관리, 143 페이지](#) 참고).
  - 단계 10** 논리적 디바이스를 생성합니다([논리적 디바이스, 149 페이지](#) 참고).
-

## 초기 구성

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

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

FXOS 새시의 단일 관리 포트에는 IPv4 주소, 게이트웨이 및 서브넷 마스크 하나만, 또는 IPv6 주소, 게이트웨이 및 네트워크 접두사 하나만 지정해야 합니다. 관리 포트 IP 주소로 IPv4 또는 IPv6 주소 중 하나를 구성할 수 있습니다.

시작하기 전에

**1** FXOS 새시에서 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 가이드](#)를 참고하십시오.

**2** 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)에서 콘솔 매개변수가 다음과 같은지 확인합니다

- 9600보드
- 8 데이터 비트
- 패리티 없음
- 1 스톱 비트

절차

**단계 1** 콘솔 포트에 연결합니다.

**단계 2** FXOS 새시의 전원을 켭니다.

FXOS 새시가 부팅할 때 자체 전원 테스트 메시지를 확인할 수 있습니다.

**단계 3** 구성되지 않은 시스템을 부팅할 경우, 설정 마법사에 시스템을 구성하는 데 필요한 다음 정보를 묻는 프롬프트가 표시됩니다.

- 설정 모드(전체 시스템 백업에서 복원 또는 초기 설정)
- 강력한 비밀번호 시행 정책(강력한 비밀번호 지침에 대해서는 [사용자 계정, 28 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**를 입력하여 설정을 저장하고 적용하거나 **no**를 입력하여 설정 마법사를 통해 일부 설정을 변경합니다.

설정 마법사를 다시 사용하도록 선택하는 경우 이전에 입력한 값이 괄호로 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)에서 콘솔 매개변수가 다음과 같은지 확인합니다.

- 9600보드
- 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 새시의 IP 주소 또는 호스트 이름을 알아야 합니다.

다음 구문 예시 중에서 하나를 사용하여 SSH, 텔넷 또는 Putty를 통해 로그인할 수 있습니다.



참고 SSH 로그인에서는 대/소문자를 구분합니다.

SSH를 사용하는 Linux 터미널에서

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

```
ssh ucs-example\jsmith@192.0.20.11
ssh ucs-example\jsmith@2001::1
```
- **ssh -lucs-auth-domain**\username {UCSM-ip-address|UCSM-ipv6-address|UCSM-host-name}
 

```
ssh -l ucs-example\jsmith 192.0.20.11
ssh -l ucs-example\jsmith 2001::1
```
- **ssh** {UCSM-ip-address|UCSM-ipv6-address|UCSM-host-name} **-lucs-auth-domain**\username
 

```
ssh 192.0.20.11 -l ucs-example\jsmith
ssh 2001::1 -l ucs-example\jsmith
```
- **sshucs-auth-domain**\username@{UCSM-ip-address|UCSM-ipv6-address}
 

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

텔넷을 사용하는 Linux 터미널에서



참고 텔넷은 기본적으로 비활성화되어 있습니다. 텔넷 활성화에 대한 지침은 [텔넷 구성, 109 페이지](#)를 참고하십시오.

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

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

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

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

Putty 클라이언트에서

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

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




---

**참고** 기본 인증이 로컬로 설정되었고 콘솔 인증이 LDAP으로 설정된 경우 Putty 클라이언트에서 **ucs-local\admin**을 사용하여 패브릭 인터커넥트에 로그인할 수 있습니다. 여기서 **admin**은 로컬 계정의 이름입니다.

---







## 라이선스 관리

Cisco 스마트 소프트웨어 라이선싱에서는 중앙 집중식으로 라이선스 풀을 구매하여 관리할 수 있습니다. 각 유닛의 라이선스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. 또한 스마트 소프트웨어 라이선싱에서는 라이선스 사용량 및 필요량을 한눈에 볼 수 있습니다.

- [Smart Software Licensing 정보](#), 15 페이지
- [Smart Software Licensing 사전 요구 사항](#), 18 페이지
- [Smart Software Licensing의 기본값](#), 18 페이지
- [Smart Software Licensing 구성](#), 19 페이지
- [FXOS 새시의 Smart License Manager Satellite](#), 21 페이지
- [영구 라이선스 예약](#), 23 페이지
- [Smart Software Licensing 모니터링](#), 24 페이지
- [Smart Software Licensing 기록](#), 25 페이지

## Smart Software Licensing 정보

이 섹션에서는 Smart Software Licensing이 적용되는 방법에 대해 설명합니다.



참고

이 섹션은 FXOS 새시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 환경 설정 가이드를 참조하십시오.

## FXOS 새시의 애플리케이션용 Smart Software Licensing

FXOS 새시의 애플리케이션의 경우, Smart Software Licensing 컨피그레이션은 FXOS 새시 관리자(Supervisor)와 애플리케이션으로 나뉩니다.

- FXOS 새시 — 관리자(Supervisor)에 모든 Smart Software Licensing 인프라를 구성하며 여기에는 License Authority와 통신하는 데 필요한 매개변수가 포함됩니다. FXOS 새시 자체는 작동하기 위한 라이선스가 필요하지 않습니다.

오프라인 라이선싱의 경우, 로컬 네트워크에 액세스할 수 있으며 온디맨드 또는 설정한 예약 기준으로 License Authority와 동기화되는 Cisco Smart Software Manager Satellite 서버를 구축할 수 있습니다.

FXOS 새시에서 지원되는 세 가지 Smart License 모델은 Smart License 기본 모드, Proxy 모드 및 Smart License Satellite 모드입니다. 새시 간 클러스터링에서는 클러스터의 각 FXOS 새시에서 동일한 Smart Licensing 모드를 활성화해야 합니다.

- 애플리케이션 — 애플리케이션의 모든 라이선스 엔타이틀먼트를 구성합니다.

## Smart Software Manager 및 어카운트

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

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

Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.



참고

아직 어카운트가 없는 경우 [새 어카운트 설정](#) 링크를 클릭합니다. Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.

기본적으로 마스터 계정의 기본 가상 계정에 라이선스가 지정됩니다. 계정 관리자는 선택적으로 추가 가상 계정을 만들 수 있습니다. 이를테면 지역, 부서, 자회사를 위한 계정을 만들 수 있습니다. 여러 가상 계정이 있으면 많은 수의 라이선스 및 디바이스를 더 편리하게 관리할 수 있습니다.

## 가상 어카운트별로 관리되는 라이선스 및 디바이스

라이선스 및 디바이스는 가상 어카운트별로 관리됩니다. 가상 계정의 디바이스에서만 해당 계정에 지정된 라이선스를 사용할 수 있습니다. 추가 라이선스가 필요할 경우 다른 가상 계정의 미사용 라이선스를 이전할 수 있습니다. 또한 가상 어카운트 간에 디바이스를 이전할 수도 있습니다.

FXOS 새시만 디바이스로 등록하는 반면 새시의 애플리케이션은 고유한 라이선스를 요청합니다. 예를 들어, 보안 모듈이 3개 있는 Firepower 9300 새시의 경우 새시는 디바이스 1개로 간주되지만 모듈은 개별 라이선스 3개를 사용합니다.

## 디바이스 등록 및 토큰

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



**참고** 디바이스 등록은 보안 모듈이 아닌 FXOS 새시 관리자(Supervisor)에서 구성됩니다.

구축 이후 시작할 때 또는 기존 디바이스에서 이 매개변수를 수동으로 구성한 후에 디바이스가 Cisco License Authority에 등록됩니다. 디바이스를 토큰과 함께 등록하면 License Authority는 디바이스와 License Authority 간의 통신을 위해 ID 인증서를 발급합니다. 이 인증서는 6개월마다 갱신되지만 1년간 유효합니다.

## License Authority와의 정기적인 통신

디바이스는 30일마다 License Authority와 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.

선택 사항으로 HTTP 프록시를 구성할 수 있습니다.

최소 90일마다 FXOS 새시가 직접 또는 HTTP 프록시를 통해 인터넷에 연결되어야 합니다. 일반 라이선스 통신은 30일마다 이루어지지만, 유예 기간이 있으므로 디바이스는 최대 90일간 콜 홈 없이 작동할 수 있습니다. 유예 기간이 지난 후 Licensing Authority에 연락해야 합니다. 아니면 특별 라이선스가 필요한 기능의 컨피그레이션을 변경할 수 없습니다. 이를 제외하면 작동에 영향을 미치지 않습니다.



**참고** 오프라인 라이선싱은 지원되지 않습니다.

## 규정 위반 상태

디바이스는 다음과 같은 상황에서 규정 위반이 될 수 있습니다.

- 과다 사용 — 디바이스에서 사용 불가능한 라이선스를 사용할 경우.
- 라이선스 만료 — 한시적인 라이선스가 만료된 경우.
- 통신 부재 — 디바이스에서 권한 재부여를 위해 Licensing Authority에 연결하지 못한 경우.

어카운트가 규정 위반 상태인지 또는 규정 위반 상태에 근접한지를 확인하려면, FXOS 새시에서 현재 사용 중인 엔타이틀먼트와 스마트 어카운트의 엔타이틀먼트를 비교해야 합니다.

규정 위반 상태에서 디바이스는 애플리케이션에 따라 일부 제한됩니다.

## Smart Call Home 인프라

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

Smart Software Licensing의 Smart Call Home을 비활성화할 수 없습니다.

## Smart Software Licensing 사전 요구 사항

- Cisco Smart Software Manager에서 마스터 계정을 만듭니다.

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

아직 어카운트가 없는 경우 [새 어카운트 설정](#) 링크를 클릭합니다. Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.

- Cisco Software Central에서 라이선스를 하나 이상 구매합니다.
- 디바이스에서 Licensing Authority와 통신할 수 있도록 디바이스에서 인터넷 액세스 또는 HTTP 프록시 액세스를 보장합니다. 오프라인 라이선싱은 지원되지 않습니다.
- 디바이스에서 Licensing Authority 서버의 이름을 확인할 수 있도록 DNS 서버를 구성합니다.
- 디바이스의 클록을 설정합니다.
- 이 섹션은 FXOS 새시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 환경 설정 가이드를 참조하십시오.

## Smart Software Licensing의 기본값

FXOS 새시 기본 컨피그레이션은 Smart Call Home 프로파일인 "SLProf"를 포함하며, 여기에서 Licensing Authority의 URL을 지정합니다.

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

## Smart Software Licensing 구성

Cisco License Authority와 통신하기 위해 HTTP 프록시를 선택적으로 구성할 수 있습니다. License Authority에 등록하려면 Smart Software Licensing 계정에서 얻은 FXOS 새시에 등록 토큰 ID를 입력해야 합니다.

절차

단계 1 (선택 사항) HTTP 프록시 구성, 19 페이지.

단계 2 License Authority에 Firepower Security Appliance 등록, 20 페이지.

### (선택 사항) HTTP 프록시 구성

네트워크에서 인터넷 액세스에 HTTP 프록시를 사용할 경우 스마트 소프트웨어 라이선싱에 대해 프록시 주소를 구성해야 합니다. 일반적으로 이 프록시는 Smart Call Home에도 사용됩니다.



참고 인증이 있는 HTTP 프록시는 지원되지 않습니다.

절차

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

예제:

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

단계 2 프록시 URL을 설정합니다.

**set http-proxy-server-urlurl**

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

예제:

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

단계 3 포트를 설정합니다.

**set http-proxy-server-portport**

예제:

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

- 단계 4 버퍼를 커밋합니다.  
**commit-buffer**
- 

## (선택 사항) Call Home URL 삭제

앞에서 구성한 Call Home URL을 삭제하려면 다음 절차를 사용하십시오.

절차

---

- 단계 1 모니터링 범위를 입력합니다.  
**scope monitoring**
- 단계 2 callhome 범위를 입력합니다.  
**scope callhome**
- 단계 3 SLProfile을 찾습니다.  
**scope profile SLProfile**
- 단계 4 목적지를 표시합니다.  
**show destination**

예제:

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

- 단계 5 URL을 삭제합니다.  
**delete destination SLDest**
- 단계 6 버퍼를 커밋합니다.  
**commit-buffer**
- 

## License Authority에 Firepower Security Appliance 등록

FXOS 새시를 등록할 때 License Authority에서는 FXOS 새시와 License Authority의 통신을 위해 ID 인증서를 발급합니다. 또한 FXOS 새시를 적절한 가상 계정에 지정합니다. 일반적으로 이 절차는 1회 수행됩니다. 그러나 이를테면 통신 문제 때문에 ID 인증서가 만료되면 나중에 FXOS 새시를 다시 등록해야 할 수 있습니다.

## 절차

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

**단계 2** FXOS 새시에 등록 토큰을 입력합니다.  
**scope license register idtoken id-token**

예제:

```
scope license
  register idtoken ZGFmNWM5NjgtYmNjYS00ZWl3L
WE3NGItMWJkOGExZjIxNGQ0LTE0NjI2NDYx%0AMDIZNT
V8N3R0dXMlZ0NjWkdR214eFZhMldBOS9CVnNEYnVKMl
g3R3dvemRD%0AY29NQT0%3D%0A
```

**단계 3** 이후에 디바이스의 등록을 취소하려면 다음을 입력합니다.  
**등록 취소**

FXOS 새시의 등록을 취소하면 어카운트에서 해당 디바이스가 제거됩니다. 디바이스의 모든 라이선스 엔타이틀먼트 및 인증서가 제거됩니다. 새 FXOS 새시의 라이선스를 확보하기 위해 등록을 취소하는 경우가 있습니다. 또는 Smart Software Manager에서 해당 디바이스를 제거할 수 있습니다.

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

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

## FXOS 새시의 Smart License Manager Satellite

Cisco Smart Software Manager Satellite는 Cisco SSM(Smart Software Manager)과 함께 작동하는 Cisco Smart Licensing의 구성 요소입니다. Cisco Smart Software Manager Satellite는 고객이 구매하고 소비한 Cisco 라이선스에 대해 실시간에 가까운 가시성과 보고 기능을 제공하여 고객 제품 라이선스를 능동적으로 관리하는 데 도움을 줍니다.

정책 또는 네트워크 연결 이유로 인해 software.cisco.com에 있는 Cisco Smart Software Manager를 직접 사용하여 Cisco 제품을 관리하지 않으려는 고객은 Cisco Smart Software Manager Satellite 온프레미스를 설치할 수 있습니다. Satellite가 활성화된 경우 FXOS 새시에서는 Smart Software Manager Satellite를 사내에 호스팅된 Cisco Smart Software Manager의 복제본으로 가정하고 보고서 라이선스 소비량을 전송합니다.

Satellite 애플리케이션을 다운로드하고 구축하면 인터넷을 사용하여 Cisco SSM에 데이터를 전송하지 않고 다음 기능을 수행할 수 있습니다.

- 라이선스 활성화 또는 등록

- 회사의 라이선스 보기
- 회사 엔터티 간 라이선스 양도

자세한 내용은 [Smart Account Manager Satellite](#)의 Smart Software Manager Satellite 설치 및 환경 설정 가이드를 참고하십시오.

## FXOS 새시의 Smart License Satellite Server 구성

다음 절차는 Smart License Manager Satellite를 사용하도록 FXOS 새시를 구성하는 방법을 보여줍니다.

시작하기 전에

- [Smart Software Licensing 사전 요구 사항](#), 18 페이지에 나열된 모든 전제 조건을 완료합니다.
- Cisco.com에서 [Smart License Satellite OVA](#) 파일을 다운로드하고 VMwareESXi 서버에 이 파일을 설치 및 구성합니다. 자세한 내용은 [Smart Software Manager Satellite 설치 가이드](#)를 참고하십시오.

절차

단계 1 Callhome 대상으로 Satellite 서버를 설정합니다.

**scopemonitoring**

**scopecall-home**

**scopeprofileSLProfile**

**scopedestinationSLDest**

**setaddress** https://ip\_address/Transportgateway/services/DeviceRequestHandler

단계 2 새 Trust Point를 생성합니다.

a) 보안 모드를 입력합니다.

**scopesecurity**

b) Trust Point를 생성하고 이름을 지정합니다.

**createtrustpoint**trustpoint\_name

c) Trust Point의 인증서 정보를 지정합니다. 참고: 인증서는 Base64 암호화 X.509(CER) 형식이어야 합니다.

**setcertchain**certchain

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

d) 커밋그레이션을 커밋합니다.

**commit-buffer**



- 단계 3 License Authority에 FXOS 새시를 등록합니다(License Authority에 Firepower Security Appliance 등록, 20 페이지 참고). Smart License Manager Satellite에서 등록 토큰을 요청하고 복사해야 합니다.

## 영구 라이선스 예약

FXOS 새시에 영구 라이선스를 할당할 수 있습니다. 이 범용 예약을 사용하면 디바이스에서 어떤 엔 타이틀먼트라도 무제한 사용할 수 있습니다.



참고 시작하기 전에 Smart Software Manager에서 사용할 수 있도록 영구 라이선스를 구매해야 합니다. 모든 계정에 대해 영구 라이선스 예약이 승인되는 것은 아닙니다. 구성을 시도하기 전에 Cisco에서 이 기능에 대한 승인을 받았는지 확인하십시오.

## 영구 라이선스 예약 구성

다음 절차는 FXOS 새시에 영구 라이선스를 할당하는 방법을 보여줍니다.

### 절차

- 단계 1 FXOS CLI에서 라이선스 예약을 활성화합니다.  
라이선스 범위  
예약 활성화
- 단계 2 라이선스 예약 범위를 지정하려면:  
라이선스 범위  
예약 범위
- 단계 3 예약 요청 코드를 생성합니다.  
범용 요청  
라이선스 예약 코드 표시
- 단계 4 Cisco Smart Software Manager 포털의 Smart Software Manager Inventory 화면으로 이동하여 Licenses(라이선스) 탭을 클릭합니다.  
<https://software.cisco.com/#SmartLicensing-Inventory>  
Licenses(라이선스) 탭에는 계정과 연결된 모든 기존 라이선스(일반 및 영구)가 표시됩니다.
- 단계 5 License Reservation(라이선스 예약)을 클릭하고, 생성된 예약 요청 코드를 상자에 입력합니다.
- 단계 6 Reserve License(라이선스 예약)를 클릭합니다.

Smart Software Manager에서 인증 코드를 생성합니다. 코드를 다운로드하거나 클립보드로 복사할 수 있습니다. 이 시점에서는 Smart Software Manager에 따라 라이선스가 사용됩니다.

**License Reservation**(라이선스 예약) 버튼이 표시되지 않으면 계정이 영구 라이선스 예약에 대해 인증되지 않은 것입니다. 이 경우 영구 라이선스 예약을 비활성화하고 일반 `smart license` 명령을 다시 입력해야 합니다.

- 단계 7 FXOS CLI에서 인증 코드를 입력합니다.  
라이선스 스마트 예약 설치 코드  
이제 FXOS에 PLR로 완전히 라이선스가 부여되었습니다.

## (선택 사항) FXOS 영구 라이선스 반환

영구 라이선스가 더 이상 필요하지 않으면 다음 절차를 사용하여 공식적으로 Smart Software Manager에 반환해야 합니다. 모든 단계를 수행하지 않으면 라이선스가 사용 중 상태로 유지되므로 다른 곳에서 사용할 수 없습니다.

### 절차

- 단계 1 FXOS CLI에서 반환 코드를 생성합니다.  
라이선스 스마트 예약 반환  
FXOS 새시의 라이선스가 즉시 취소되고 Evaluation(평가) 상태로 전환됩니다.
- 단계 2 Smart Software Manager에서 FXOS 인스턴스를 찾을 수 있도록 FXOS UDI(universal device identifier)를 확인합니다.  
라이선스 **udi** 표시
- 단계 3 Smart Software Manager Inventory 화면으로 이동하여 **Product Instances**(제품 인스턴스) 탭을 클릭합니다.  
<https://software.cisco.com/#SmartLicensing-Inventory>
- 단계 4 UDI(universal device identifier)를 사용하여 FXOS 새시를 검색합니다.
- 단계 5 **Actions**(작업) > **Remove**(제거)를 선택하고, 생성된 반환 코드를 상자에 입력합니다.
- 단계 6 **Remove Product Instance**(제품 인스턴스 제거)를 클릭합니다.  
영구 라이선스가 사용 가능한 풀로 반환됩니다.

## Smart Software Licensing 모니터링

라이선스 상태를 보려면 다음 명령을 참고하십시오.

- **show license all**

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

- **show license status**

- **show license techsupport**

## Smart Software Licensing 기록

기능 이름	플랫폼 릴리스	설명
FXOS 새시의 Cisco 스마트 소프트웨어 라이선싱	1.1(1)	<p>스마트 소프트웨어 라이선싱에서는 라이선스 풀을 구매하여 관리할 수 있습니다. 스마트 라이선스는 특정 일련 번호에 연결되지 않습니다. 각 유닛의 라이선스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. 또한 스마트 소프트웨어 라이선싱에서는 라이선스 사용량 및 필요량을 한눈에 볼 수 있습니다. Smart Software Licensing 컨피그레이션은 FXOS 새시 관리자 (Supervisor)와 보안 모듈로 나뉩니다.</p> <p>추가된 명령: <b>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</b></p>





## 사용자 관리

---

- 사용자 계정, 28 페이지
- 사용자 이름 지침, 29 페이지
- 비밀번호 지침, 29 페이지
- 원격 인증에 대한 지침, 30 페이지
- 사용자 역할, 33 페이지
- 로컬 인증 사용자에게 대한 비밀번호 프로필, 33 페이지
- 기본 인증 서비스 선택, 34 페이지
- 절대 세션 시간 초과 구성, 36 페이지
- 원격 사용자에게 대한 역할 정책 구성, 36 페이지
- 로컬로 인증된 사용자의 비밀번호 보안 수준 확인 활성화, 37 페이지
- 최대 로그인 시도 횟수 설정, 38 페이지
- 변경 간격에 대해 최대 비밀번호 변경 횟수 구성, 39 페이지
- 최소 비밀번호 길이 확인 구성, 39 페이지
- 비밀번호에 대해 변경 간격 없음 구성, 40 페이지
- 비밀번호 기록 수 구성, 41 페이지
- 로컬 사용자 계정 생성, 41 페이지
- 로컬 사용자 계정 삭제, 43 페이지
- 로컬 사용자 계정 활성화 또는 비활성화, 44 페이지
- 로컬 인증 사용자에게 대한 비밀번호 기록 지우기, 44 페이지

# 사용자 계정

사용자 계정을 사용하여 시스템에 액세스합니다. 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 수정하거나 삭제할 수 없습니다. 이 계정은 시스템 관리자 또는 슈퍼 사용자(superuser) 계정이며 전체 권한을 갖습니다. 관리자 계정에는 기본 비밀번호가 할당되지 않으므로, 초기 시스템 설치 과정에서 비밀번호를 선택해야 합니다.

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

## 로컬 인증 사용자 계정

로컬로 인증된 사용자 계정은 새시를 통해 직접 인증되며 관리자 또는 AAA 권한을 보유한 사용자에 의해 활성화 또는 비활성화될 수 있습니다. 로컬 사용자 계정이 비활성화되면 사용자가 로그인할 수 없습니다. 비활성화된 로컬 사용자 계정에 대한 컨피그레이션 세부사항은 데이터베이스에 의해 삭제되지 않습니다. 비활성화된 로컬 사용자 계정을 다시 활성화하는 경우, 어카운트는 사용자 이름 및 비밀번호를 포함한 기존 컨피그레이션으로 다시 활성화됩니다.

## 원격 인증 사용자 계정

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

사용자가 로컬 사용자 계정과 원격 사용자 계정을 동시에 유지할 경우 로컬 사용자 계정에 정의된 역할이 원격 사용자 계정의 역할을 재정의합니다.

원격 인증 지침, 그리고 원격 인증 공급자의 구성 및 삭제 방법에 대한 자세한 내용은 다음 항목을 참조하십시오.

- [원격 인증에 대한 지침, 30 페이지](#)
- [LDAP 제공자 구성, 129 페이지](#)
- [RADIUS 제공자 구성, 133 페이지](#)
- [TACACS+ 제공자 구성, 136 페이지](#)

## 사용자 계정 만료

미리 정의된 시간에 만료하도록 사용자 계정을 구성할 수 있습니다. 만료 시간이 되면 사용자 계정은 비활성화됩니다.

기본적으로 사용자 계정은 만료되지 않습니다.

사용자 계정에 만료일을 구성하면, 해당 계정을 만료하도록 재구성할 수 없습니다. 그러나 계정에 최신 만료일을 사용할 수 있도록 구성할 수는 있습니다.

## 사용자 이름 지침

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

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

## 비밀번호 지침

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

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

- 8자 이상, 80자 이하여야 합니다.



참고 Common Criteria 요구 사항을 준수하기 위해 시스템에서 최소 15자 비밀번호 길이를 선택적으로 구성할 수 있습니다. 자세한 내용은 [최소 비밀번호 길이 확인 구성, 39 페이지](#)를 참고하십시오.

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

- 하나 이상의 영숫자 외 문자(특수 문자)를 포함해야 합니다.
- aaabbb와 같이 한 문자가 3번 이상 연속적으로 나와서는 안 됩니다.
- 어떤 순서로든 3개의 연속 숫자 또는 문자를 포함해서는 안 됩니다(예: passwordABC 또는 password321).
- 사용자 이름 또는 사용자 이름의 역순과 같아서는 안 됩니다.
- 비밀번호 사전 검사를 통과해야 합니다. 예를 들어 비밀번호가 표준 사전 단어를 기반으로 해서 는 안 됩니다.
- \$(달러 기호), ?(물음표), =(등호) 기호를 포함해서는 안 됩니다.
- 로컬 사용자 및 관리자 어카운트 비밀번호는 비어 있지 않아야 합니다.

## 원격 인증에 대한 지침

지원되는 원격 인증 서비스 중 하나가 시스템에 구성될 경우, FXOS 새시에서 시스템과 통신할 수 있도록 그 서비스에 대한 제공자를 생성해야 합니다. 다음 지침은 사용자 인증에 영향을 미칩니다.

### 원격 인증 서비스의 사용자 계정

사용자 계정은 FXOS 새시의 로컬에 두거나 원격 인증 서버에 둘 수 있습니다.

FXOS Chassis Manager GUI 및 FXOS CLI에서 원격 인증 서비스를 통해 로그인한 사용자의 임시 세션을 볼 수 있습니다.

### 원격 인증 서비스의 사용자 역할

원격 인증 서버에 사용자 계정을 생성할 경우 그 계정은 FXOS 새시에서 작업하는 데 필요한 역할을 포함하고 그 역할의 이름이 FXOS에서 사용되는 이름과 일치해야 합니다. 역할 정책에 따라 사용자가 로그인하지 못하거나 읽기 전용 권한만 가질 수도 있습니다.

### 원격 인증 제공자의 사용자 특성

RADIUS 및 TACAS+ 컨피그레이션에서는 사용자가 Firepower Chassis Manager 및 FXOS CLI에 로그인하는 원격 인증 제공자 각각에서 FXOS 새시에 대한 사용자 특성을 구성해야 합니다. 이 사용자 특성은 각 사용자에게 지정된 역할 및 로컬을 저장합니다.

사용자가 로그인하면 FXOS에서 다음을 수행합니다.

- 1 원격 인증 서비스에 쿼리합니다.
- 2 사용자를 검증합니다.
- 3 사용자가 검증되면 해당 사용자에게 할당된 역할 및 로케일을 확인합니다.

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



인증 제공자	커스텀 특성	스키마 확장	특성 ID 요구 사항
LDAP	선택 사항	<p>다음 중 하나를 선택하여 수행할 수 있습니다.</p> <ul style="list-style-type: none"> <li>LDAP 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 특성을 구성합니다.</li> <li>LDAP 스키마를 확장하고 CiscoAVPair와 같은 고유한 이름으로 커스텀 특성을 생성합니다.</li> </ul>	<p>Cisco LDAP 구현에서는 유니코드 형식의 특성이 필요합니다.</p> <p>CiscoAVPair 커스텀 특성을 생성하려는 경우 특성 ID로 1.3.6.1.4.1.9.287247.1을 사용합니다.</p> <p>샘플 OID가 다음 섹션에 나와 있습니다.</p>
RADIUS	선택 사항	<p>다음 중 하나를 선택하여 수행할 수 있습니다.</p> <ul style="list-style-type: none"> <li>RADIUS 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 특성을 사용합니다.</li> <li>RADIUS 스키마를 확장하고 cisco-avpair와 같은 고유한 이름으로 커스텀 특성을 생성합니다.</li> </ul>	<p>Cisco RADIUS 구현의 벤더 ID는 009, 특성의 벤더 ID는 001입니다.</p> <p>다음 구문의 예에서는 cisco-avpair 특성을 생성하려는 경우 여러 사용자 역할 및 로케일을 지정하는 방법을 보여줍니다.</p> <pre>shell:roles="admin,aaa" shell:locales="L1,abc".</pre> <p>여러 값을 구분하는 기호로 쉼표 ","를 사용합니다.</p>

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

**LDAP 사용자 특성에 대한 샘플 OID**

다음은 커스텀 `CiscoAVPair` 특성에 대한 샘플 OID입니다.

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

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

## 사용자 역할

시스템에는 다음과 같은 사용자 역할이 포함됩니다.

### 관리자

전체 시스템에 대한 완전한 읽기 및 쓰기 액세스. 기본 관리자 계정은 기본적으로 이 역할을 지정받으며 변경할 수 없습니다.

### 읽기 전용

시스템 상태를 수정할 권한이 없는, 시스템 컨피그레이션에 대한 읽기 전용 액세스.

### 운영

NTP 컨피그레이션, Smart Licensing에 대한 Smart Call Home 컨피그레이션, 시스템 로그(syslog 서버 및 장애 포함)에 대한 읽기 및 쓰기 액세스. 나머지 시스템에 대한 읽기 액세스 권한입니다.

### AAA 관리자

사용자, 역할, AAA 컨피그레이션에 대한 읽기-쓰기 액세스 권한입니다. 나머지 시스템에 대한 읽기 액세스 권한입니다.

## 로컬 인증 사용자에게 대한 비밀번호 프로필

비밀번호 프로파일에는 모든 로컬로 인증된 사용자에게 대한 비밀번호 기록 및 비밀번호 변경 간격 속성이 포함되어 있습니다. 로컬에서 인증된 각 사용자에게는 다른 비밀번호 프로필을 지정할 수 없습니다.

### 비밀번호 기록 횟수

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

비밀번호를 재사용하려면 사용자는 우선 비밀번호 기록 횟수에 구성되는 비밀번호의 수를 만들고 사용해야 합니다. 예를 들어 비밀번호 기록 횟수를 8로 설정한 경우, 로컬에서 인증된 사용자는 9번째 비밀번호가 만료될 때까지 첫 번째 비밀번호를 재사용할 수 없습니다.

기본적으로 비밀번호 기록은 0으로 설정됩니다. 이 값은 기록 횟수를 비활성화하며 사용자가 이전 비밀번호를 언제든지 재사용할 수 있도록 합니다.

필요한 경우, 로컬에서 인증된 사용자의 비밀번호 기록 횟수를 지우고 이전 비밀번호 재사용을 활성화할 수 있습니다.

비밀번호 변경 간격

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

간격 컨피그레이션	설명	예
No password change allowed(비밀번호 변경 허용 안 됨)	이 옵션은 로컬에서 인증된 사용자가 비밀번호 변경 후 지정된 시간 이내에는 비밀번호를 변경할 수 없도록 합니다.  변경 없음 간격을 1~745시간으로 지정할 수 있습니다. 기본적으로 변경 없음 간격은 24시간입니다.	예를 들어 로컬에서 인증된 사용자가 비밀번호를 변경한 후 48시간 이내에 비밀번호를 변경하지 못하도록 하려면 다음을 설정합니다.  • 간격 동안 변경 비활성화 • 변경 불가 간격 - 48
Password changes allowed within change interval(변경 간격 내에 비밀번호 변경 허용됨)	이 옵션은 로컬에서 인증된 사용자가 사전 정의된 간격 내에 비밀번호를 변경할 수 있는 최대 횟수를 지정합니다.  변경 간격을 1~745시간으로 지정하고 최대 비밀번호 변경 횟수를 0~10으로 지정할 수 있습니다. 기본적으로 로컬에서 인증된 사용자는 48시간 간격 내에 비밀번호를 최대 2번 변경하는 것이 허용됩니다.	예를 들어, 로컬로 인증된 사용자가 비밀번호를 변경한 후 24시간 이내에 비밀번호를 최대 한 번 변경하도록 허용하려면 다음을 설정합니다.  • 변경 지속 간격 - 활성화 • 변경 수 - 1 • 변경 간격 - 24

## 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 로그인하도록 허용
- **radius** — RADIUS 인증 지정
- **tacacs** — TACACS+ 인증 지정

단계 4 (선택 사항) 해당하는 경우, 연결된 제공자 그룹을 지정합니다.

```
Firepower-chassis /security/default-auth # set auth-server-group auth-serv-group-name
```

단계 5 (선택 사항) 이 도메인에 있는 사용자에게 대한 새로고침 요청 사이에 허용되는 최대 시간을 지정합니다.

```
Firepower-chassis /security/default-auth # set refresh-period seconds
```

0~600의 정수를 지정합니다. 기본값은 600초입니다.

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

단계 6 (선택 사항) FXOS에서 웹 세션이 종료되었다고 간주하기 전 마지막 새로고침 요청 이후에 경과한 최대 시간을 지정합니다.

```
Firepower-chassis /security/default-auth # set session-timeout seconds
```

0~600의 정수를 지정합니다. 기본값은 600초입니다.

참고 RADIUS 또는 TACACS+ 영역에 대한 2단계 인증을 설정한 경우, **session-refresh** 및 **session-timeout** 간격을 늘려 원격 사용자가 빈번하게 재인증하지 않아도 되도록 설정하는 것을 고려해 보십시오.

단계 7 (선택 사항) 영역에 대한 2단계 인증 방법을 설정합니다.

```
Firepower-chassis /security/default-auth # set use-2-factor yes
```

참고 2단계 인증은 RADIUS 및 TACACS+ 영역에만 적용됩니다.

단계 8 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
commit-buffer
```

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

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

## 절대 세션 시간 초과 구성

FXOS 새시에는 세션 사용과 상관없이 절대 세션 시간 초과 기간이 지나면 사용자 세션을 닫는 절대 세션 시간 초과 설정이 있습니다. 이 절대 시간 초과 기능은 시리얼 콘솔, SSH, HTTPS를 비롯한 모든 액세스 형식에서 전역적으로 적용됩니다.

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

### 절차

- 단계 1 보안 모드를 입력합니다.  
Firepower-chassis # **scope security**
- 단계 2 기본 권한 부여 보안 모드를 입력합니다.  
Firepower-chassis /security # **scopedefault-auth**
- 단계 3 절대 세션 시간 초과를 설정합니다.  
Firepower-chassis /security/default-auth # **set absolute-session-timeoutseconds**
- 단계 4 (선택 사항) 세션 및 절대 세션 시간 초과 설정을 봅니다.  
Firepower-chassis /security/default-auth # **show detail**

### 예제:

```
Default authentication(기본 인증):
Admin Realm(관리자 영역): Local(로컬)
Operational Realm(운영 영역): Local(로컬)
Web session refresh period(in secs)(웹 세션 새로 고침 기간(초)): 600
Session timeout(in secs) for web, ssh, telnet sessions(웹, ssh, 텔넷 세션에 대한 세션 시간 초과(초)): 600
Absolute Session timeout(in secs) for web, ssh, telnet sessions(웹, ssh, 텔넷 세션에 대한 절대 세션 시간 초과(초)): 3600
Serial Console Session timeout(in secs)(시리얼 콘솔 세션 시간 초과(초)): 600
Serial Console Absolute Session timeout(in secs)(시리얼 콘솔 절대 세션 시간 초과(초)): 3600
Admin Authentication server group(관리자 인증 서버 그룹):
Operational Authentication server group(운영 인증 서버 그룹):
Use of 2nd factor(두 번째 요인 사용): No(아니요)
```

## 원격 사용자에게 대한 역할 정책 구성

기본적으로 LDAP, RADIUS 또는 TACACS 프로토콜을 사용하여 원격 서버에서 Firepower Chassis Manager 또는 FXOS CLI에 로그인하는 모든 사용자에게 읽기 전용 액세스 권한이 부여됩니다. 보안

상의 이유로, 설정된 사용자 역할과 일치하는 사용자로 액세스를 제한하는 것이 바람직할 수 있습니다.

원격 사용자에 대한 역할 정책을 다음과 같이 구성할 수 있습니다.

**assign-default-role**

사용자가 로그인을 시도하고 원격 인증 제공자가 인증 정보와 함께 사용자 역할을 제공하지 않는 경우, 사용자는 읽기 전용 사용자 역할로 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 # scope security
```

단계 2 Firepower Chassis Manager 및 FXOS CLI에 대한 사용자 액세스가 사용자 역할을 기준으로 제한되어야 하는지를 지정합니다.

```
Firepower-chassis /security # set remote-user default-role {assign-default-role | no-login}
```

단계 3 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
Firepower-chassis /security # commit-buffer
```

다음 예에서는 원격 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 #
```

## 로컬로 인증된 사용자의 비밀번호 보안 수준 확인 활성화

비밀번호 보안 수준 확인이 활성화된 경우에는 Firepower eXtensible 운영 체제에서 사용자가 강력한 비밀번호 지침을 따르지 않는 비밀번호를 선택하도록 허용하지 않습니다(비밀번호 지침, 29 페이지 참고).

절차

단계 1 보안 모드를 입력합니다.

```
Firepower-chassis # scope security
```

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 #
```

## 최대 로그인 시도 횟수 설정

허용된 최대 횟수만큼 로그인 시도에 실패하면 지정된 시간 동안 사용자가 잠기도록 FXOS 재시를 구성할 수 있습니다. 설정된 로그인 최대 시도 횟수를 초과하면 사용자가 시스템에서 잠깁니다. 사용자가 잠겼음을 나타내는 알림이 표시되지 않습니다. 이 경우 사용자는 다시 로그인을 시도하려면 지정된 시간 동안 기다려야 합니다.

최대 로그인 시도 횟수를 구성하려면 다음 단계를 수행하십시오.



참고

- 최대 로그인 시도 횟수를 초과하면 모든 유형의 사용자 계정(관리자 포함)이 시스템에서 잠깁니다.

이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 [보안 인증 컴플라이언스, 57 페이지](#)를 참고하십시오.

절차

- 
- 단계 1** FXOS CLI에서 보안 모드로 들어갑니다.
- ```
scopesystem
scopesecurity
```
- 단계 2** 최대 로그인 시도 실패 횟수를 설정합니다.
- ```
setmax-login-attempts
```
- max\_login*  
*max\_login* 값은 0~10의 정수입니다.
- 단계 3** 최대 로그인 시도 횟수에 도달한 후 사용자가 시스템에서 잠긴 상태로 유지되는 시간(초)을 지정합니다.
- ```
setuser-account-unlock-time
```
- unlock\_time*
- 단계 4** 컨피그레이션을 커밋합니다.
- ```
commit-buffer
```
-



# 변경 간격에 대해 최대 비밀번호 변경 횟수 구성

## 절차

- 단계 1** 보안 모드를 입력합니다.  
`Firepower-chassis # scope security`
- 단계 2** 비밀번호 프로파일 보안 모드를 입력합니다.  
`Firepower-chassis /security # scope password-profile`
- 단계 3** 로컬로 인증된 사용자가 지정된 시간 이내에 변경할 수 있는 비밀번호 변경 수를 제한합니다.  
`Firepower-chassis /security/password-profile # set change-during-interval enable`
- 단계 4** 로컬로 인증된 사용자가 변경 간격 동안 비밀번호를 변경할 수 있는 최대 횟수를 지정합니다.  
`Firepower-chassis /security/password-profile # set change-count pass-change-num`  
 0 ~ 10의 어떤 값이든 가능합니다.
- 단계 5** **Change Count**(변경 횟수) 필드에 지정된 비밀번호 변경 횟수가 적용되는 최대 시간을 지정합니다.  
`Firepower-chassis /security/password-profile # set change-interval num-of-hours`  
 1시간 ~ 745시간의 어떤 값이든 가능합니다.  
 예를 들어, 이 필드가 48로 설정되고 **Change Count**(변경 횟수) 필드가 2로 설정된 경우 로컬로 인증된 사용자는 48시간 이내에 비밀번호를 최대 2번 변경할 수 있습니다.
- 단계 6** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
`Firepower-chassis /security/password-profile # commit-buffer`

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

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

# 최소 비밀번호 길이 확인 구성

최소 비밀번호 길이 확인을 활성화하는 경우 지정된 최소 문자 수의 비밀번호를 만들어야 합니다. 예를 들어 `min_length` 옵션이 15로 설정된 경우 15자 이상을 사용해 비밀번호를 만들어야 합니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 허용하는 숫자 중 하나입니다. 자세한 내용은 [보안 인증 컴플라이언스, 57 페이지](#)를 참고하십시오.

최소 비밀번호 길이 확인을 구성하려면 다음 단계를 수행하십시오.

## 절차

- 
- 단계 1 FXOS CLI에서 보안 모드로 들어갑니다.
  - 단계 2 **scopesystem**  
**scopesecurity**
  - 단계 3 비밀번호 프로파일 보안 모드를 입력합니다.  
**scopepassword-profile**
  - 단계 4 최소 비밀번호 길이를 지정합니다.  
**setmin-password-length***min\_length*
  - 단계 5 컨피그레이션을 커밋합니다.  
**commit-buffer**
- 

## 비밀번호에 대해 변경 간격 없음 구성

## 절차

- 
- 단계 1 보안 모드를 입력합니다.  
Firepower-chassis # **scope security**
  - 단계 2 비밀번호 프로파일 보안 모드를 입력합니다.  
Firepower-chassis /security # **scope password-profile**
  - 단계 3 해당 간격 동안 변경 기능을 비활성화합니다.  
Firepower-chassis /security/password-profile # **set change-during-interval disable**
  - 단계 4 로컬로 인증된 사용자가 새로 생성된 비밀번호를 변경하기 전까지 기다려야 하는 최소 시간을 지정합니다.  
Firepower-chassis /security/password-profile # **set no-change-interval** *min-num-hours*  
1시간 ~ 745시간의 어떤 값이든 가능합니다.  
이 간격은 **Change During Interval**(사이에 변경) 속성이 **Disable**(비활성)로 설정되지 않은 경우 무시됩니다.
  - 단계 5 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
Firepower-chassis /security/password-profile # **commit-buffer**
- 

다음의 예에서는 해당 간격 동안 변경 옵션을 비활성화하고 변경 안 함 간격을 72시간으로 설정하고 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 # scope security
```

단계 2 비밀번호 프로파일 보안 모드를 입력합니다.

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

단계 3 로컬로 인증된 사용자가 이전에 사용한 비밀번호를 재사용하기 전에 생성해야 하는 고유한 비밀번호 수를 지정합니다.

```
Firepower-chassis /security/password-profile # set history-count num-of-passwords
```

0 ~ 15의 어떤 값이든 가능합니다.

기본적으로 **History Count**(기록 수) 필드는 0으로 설정되어 있습니다. 즉 기록 수를 비활성화하므로 이전에 사용한 비밀번호를 언제라도 재사용할 수 있습니다.

단계 4 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/password-profile # commit-buffer
```

다음 예에서는 비밀번호 기록 수를 구성하고 트랜잭션을 커밋합니다.

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

## 로컬 사용자 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
```

여기서 *local-user-name*은 이 계정에 로그인할 때 사용할 계정 이름입니다. 이름은 고유해야 하며 사용자 계정 이름에 대한 지침 및 제한 사항을 따라야 합니다([사용자 이름 지침, 29 페이지 참조](#)).

사용자를 생성한 후에는 로그인 ID를 변경할 수 없습니다. 사용자 계정을 삭제하고 새로 만들어야 합니다.

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*

비밀번호 보안 수준 확인을 활성화하면 사용자의 비밀번호가 더욱 강력해지며, 보안 수준 확인 요건을 충족하지 않는 비밀번호를 Firepower eXtensible 운영 체제에서 거부합니다([비밀번호 지침, 29 페이지](#) 참조).

단계 5 (선택 사항) 사용자의 이름을 지정합니다.

```
Firepower-chassis /security/local-user # set firstname first-name
```

단계 6 (선택 사항) 사용자의 성을 지정합니다.

```
Firepower-chassis /security/local-user # set lastname last-name
```

단계 7 (선택 사항) 사용자 계정이 만료되는 날짜를 지정합니다. *month* 인수는 월 이름의 처음 세 글자입니다.

```
Firepower-chassis /security/local-user # set expiration month day-of-month year
```

참고 사용자 계정에 만료일을 구성하면, 해당 계정을 만료하도록 재구성할 수 없습니다. 그러나 계정에 최신 만료일을 사용할 수 있도록 구성할 수는 있습니다.

단계 8 (선택 사항) 사용자의 이메일 주소를 지정합니다.

```
Firepower-chassis /security/local-user # set email email-addr
```

단계 9 (선택 사항) 사용자 전화 번호를 지정합니다.

```
Firepower-chassis /security/local-user # set phone phone-num
```

단계 10 (선택 사항) 비밀번호 없는 액세스에 사용되는 SSH 키를 지정합니다.

```
Firepower-chassis /security/local-user # set sshkey ssh-key
```

단계 11 모든 사용자에게 기본적으로 *read-only* 역할이 할당되며 이 역할은 제거할 수 없습니다. 사용자에게 할당할 각 추가 역할에 대해:

```
Firepower-chassis /security/local-user # create rolename
```

여기서 *role-name*은 사용자 계정에 할당하고자 하는 권한을 나타내는 역할입니다([사용자 역할, 33 페이지](#) 참조).

참고 사용자 역할 및 권한의 변경은 사용자가 다음에 로그인할 때 적용됩니다. 사용자가 로그인할 때 새 역할을 지정하거나 사용자 계정의 기존 역할을 삭제할 경우 활성 세션에서는 기존의 역할 및 권한을 유지합니다.

단계 12 할당된 역할을 사용자로부터 제거하려면:

```
Firepower-chassis /security/local-user # delete rolename
```

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

단계 13 트랜잭션을 커밋합니다.

```
Firepower-chassis security/local-user # commit-buffer
```

다음 예에서는 kikipopo라는 이름의 사용자 계정을 생성하고, 이 사용자 계정을 활성화하며, 비밀번호를 foo12345로 설정하고, 관리자 사용자 역할을 할당하고, 트랜잭션을 커밋합니다.

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

다음 예에서는 lincey라는 이름의 사용자 계정을 생성하고, 이 사용자 계정을 활성화하며, 비밀번호 없는 액세스에 사용되는 OpenSSH 키를 설정하고, aaa 및 운영 사용자 역할을 할당하고, 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/S1f30k1CWjnV3lgdXMzO0WU15iPw851kdQqap+NFuNmHcb4K
iaQB8X/PDdmt1xQQcawclj+k8f4VcOelBxlsGk5luq51slob1VOIEwckEL/h51rdbN1I8y3SS9I/gGiBZ9ARlop9LDpD
m8HPh2LOgyH7Ei1MI8="
Firepower-chassis /security/local-user* # create role aaa
Firepower-chassis /security/local-user* # create role operations
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

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/S1f30k1CWjnV3lgdXMzO0WU15iPw8
>51kdQqap+NFuNmHcb4KiaQB8X/PDdmt1xQQcawclj+k8f4VcOelBxlsGk5luq51slob1VO
>IEwckEL/h51rdbN1I8y3SS9I/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 활성화하거나 비활성화할 사용자의 로컬 사용자 보안 모드를 입력합니다.

```
Firepower-chassis /security # scope local-user local-user-name
```

단계 3 로컬 사용자 계정을 활성화할지 또는 비활성화할지를 지정합니다.

```
Firepower-chassis /security/local-user # set account-status {active | inactive}
```

참고 관리자 사용자 계정은 항상 활성화로 설정됩니다. 이는 수정할 수 없습니다.

다음 예에서는 어카운팅이라고 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
```

## 로컬 인증 사용자에 대한 비밀번호 기록 지우기

절차

단계 1 보안 모드를 입력합니다.

```
Firepower-chassis # scope security
```

단계 2 지정된 사용자 계정에 대한 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 #
```







## 이미지 관리

- [이미지 관리 정보, 47 페이지](#)
- [Cisco.com에서 이미지 다운로드, 48 페이지](#)
- [Firepower eXtensible 운영 체제 소프트웨어 이미지를 FXOS 새시에 다운로드, 48 페이지](#)
- [이미지의 무결성 확인, 49 페이지](#)
- [Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드, 50 페이지](#)
- [FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드, 50 페이지](#)
- [논리적 디바이스를 위한 이미지 버전 업데이트, 53 페이지](#)
- [펌웨어 업그레이드, 54 페이지](#)

## 이미지 관리 정보

FXOS 새시는 다음의 2가지 기본 이미지 유형을 사용합니다.



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

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



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

## Cisco.com에서 이미지 다운로드

시작하기 전에

Cisco.com 어카운트가 있어야 합니다.

절차

- 
- 단계 1 웹 브라우저를 사용하여 <http://www.cisco.com/go/firepower9300-software> 또는 <http://www.cisco.com/go/firepower4100-software>로 이동합니다.  
FXOS 새시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.
- 단계 2 적절한 소프트웨어 이미지를 찾은 다음 로컬 컴퓨터에 다운로드합니다.
- 

## Firepower eXtensible 운영 체제 소프트웨어 이미지를 FXOS 새시에 다운로드

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

시작하기 전에

컨피그레이션 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

- 이미지를 복사하고 있는 원본 서버의 IP 주소 및 인증 자격 증명
- FXOS 이미지 파일의 정규화된 이름

절차

- 
- 단계 1 펌웨어 모드를 입력합니다.  
Firepower-chassis # **scope firmware**
- 단계 2 FXOS 소프트웨어 이미지를 다운로드합니다.  
Firepower-chassis /firmware # **download image URL**  
다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

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

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

다음 예에서는 SCP 프로토콜을 사용하여 이미지를 복사합니다.

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

## 이미지의 무결성 확인

FXOS 새시에 새 이미지가 추가되면 이미지의 무결성이 자동으로 확인됩니다. 필요한 경우 다음 절차를 사용하여 이미지의 무결성을 수동으로 확인할 수 있습니다.

### 절차

- 단계 1 FXOS CLI에 연결합니다([FXOS CLI 액세스](#), 12 페이지 참고).
- 단계 2 펌웨어 모드를 입력합니다.  
**Firepower-chassis# scope firmware**
- 단계 3 이미지를 나열합니다.  
**Firepower-chassis /firmware # showpackage**
- 단계 4 이미지를 확인합니다.  
**Firepower-chassis /firmware # verifyplatform-packversionversion\_number**  
*version\_number*는 확인 중인 FXOS 플랫폼 버전의 버전 번호입니다(예: 1.1(2.51)).
- 단계 5 확인하는 데 몇 분 정도 걸릴 수 있다는 메시지가 표시됩니다.  
**yes**를 입력하여 검증을 계속할 것인지 확인합니다.
- 단계 6 이미지 확인 상태를 점검하려면:  
**Firepower-chassis /firmware # showvalidate-task**

## Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드

시작하기 전에

Cisco.com에서 플랫폼 번들 소프트웨어 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 48 페이지 참고)한 다음 해당 이미지를 FXOS 새시에 다운로드합니다(FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드, 50 페이지 참고).

절차

- 단계 1 FXOS CLI에 연결합니다(FXOS CLI 액세스, 12 페이지 참고).
- 단계 2 펌웨어 모드를 입력합니다.  
Firepower-chassis# **scope firmware**
- 단계 3 자동 설치 모드를 입력합니다.  
Firepower-chassis /firmware # **scope auto-install**
- 단계 4 FXOS 플랫폼 번들을 설치합니다.  
Firepower-chassis /firmware/auto-install # **install platform platform-vers version\_number**  
*version\_number*는 설치 중인 FXOS 플랫폼 번들의 버전 번호입니다(예: 1.1(2.51)).
- 단계 5 시스템은 설치할 소프트웨어 패키지를 먼저 확인합니다. 시스템은 현재 설치된 애플리케이션과 지정된 FXOS 플랫폼 소프트웨어 패키지 간에 비호환성이 있는지 알려줍니다. 또한 기존 세션이 종료되고 시스템이 업그레이드의 일부로 재부팅되어야 한다고 경고합니다.  
**yes**를 입력하여 검증을 계속할 것인지 확인합니다.
- 단계 6 **yes**를 입력하여 설치를 계속할 것인지 확인하거나 **no**를 입력하여 설치를 취소합니다.  
Firepower eXtensible 운영 체제에서는 번들의 압축을 풀고 구성 요소를 업그레이드하거나 다시 로드합니다.
- 단계 7 업그레이드 프로세스를 모니터링하려면 다음과 같이 합니다.
  - a) **scope firmware**를 입력합니다.
  - b) **scope auto-install**을 입력합니다.
  - c) **show fsm status expand**를 입력합니다.

## FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드

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

시작하기 전에

컨피그레이션 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

- 이미지를 복사하고 있는 원본 서버의 IP 주소 및 인증 자격 증명
- 소프트웨어 이미지 파일의 정규화된 이름

절차

단계 1 보안 서비스 모드를 입력합니다.

```
Firepower-chassis # scope ssa
```

단계 2 애플리케이션 소프트웨어 모드를 입력합니다.

```
Firepower-chassis /ssa # scope app-software
```

단계 3 논리적 디바이스 소프트웨어 이미지를 다운로드합니다.

```
Firepower-chassis /ssa/app-software # download image URL
```

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

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

단계 4 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.

```
Firepower-chassis /ssa/app-software # show download-task
```

단계 5 다음 명령을 사용하여 다운로드한 애플리케이션을 확인합니다.

```
Firepower-chassis /ssa/app-software # up
```

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

단계 6 다음의 명령을 사용하여 특정 애플리케이션에 대한 세부사항을 확인합니다.

```
Firepower-chassis /ssa # scope app application_type image_version
```

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

다음 예에서는 SCP 프로토콜을 사용하여 이미지를 복사합니다.

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

Downloads for Application Software:

File Name	Protocol	Server	Userid	State
cisco-asa.9.4.1.65.csp	Scp	192.168.1.1	user	Downloaded

```
Firepower-chassis /ssa/app-software # up
```

```

Firepower-chassis /ssa # show app

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

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

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

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

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

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

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

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

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

```

# 논리적 디바이스를 위한 이미지 버전 업데이트

시작하기 전에



참고

Firepower Threat Defense 논리적 디바이스를 처음 생성한 후에는 Firepower Chassis Manager 또는 FXOS CLI를 사용하여 Firepower Threat Defense 논리적 디바이스를 업그레이드하지 않습니다. Firepower Threat Defense 논리적 디바이스를 업그레이드하려면 Firepower Management Center를 사용해야 합니다. 자세한 내용은 Firepower System 릴리스 노트를 참조하십시오. <http://www.cisco.com/c/en/us/support/security/defense-center/products-release-notes-list.html>

또한 Firepower Threat Defense 논리적 디바이스에 대한 업데이트는 Firepower Chassis Manager의 **Logical Devices**(논리적 디바이스) > **Edit**(수정) 및 **System**(시스템) > **Updates**(업데이트) 페이지에 반영되지 않습니다. 이러한 페이지에 표시되는 버전은 Firepower Threat Defense 논리적 디바이스를 만드는 데 사용된 소프트웨어 버전(CSP 이미지)을 나타냅니다.

Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 48 페이지 참고)한 다음 해당 이미지를 FXOS 새시에 다운로드합니다(FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드, 50 페이지 참고).

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

## 절차

단계 1 보안 서비스 모드를 입력합니다.

```
Firepower-chassis # scope ssa
```

단계 2 업데이트 중인 보안 모듈의 범위를 설정합니다.

```
Firepower-chassis /ssa # scope slot slot_number
```

단계 3 업데이트 중인 애플리케이션의 범위를 설정합니다.

```
Firepower-chassis /ssa/slot # scope app-instance app_template
```

단계 4 시작 버전을 업데이트하려는 버전으로 설정합니다.

```
Firepower-chassis /ssa/slot/app-instance # set startup-version version_number
```

단계 5 컨피그레이션을 커밋합니다.

```
commit-buffer
```

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

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

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

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

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

## 펌웨어 업그레이드

FXOS 새시에서 펌웨어를 업그레이드하려면 다음 절차를 사용하십시오.

### 절차

- 단계 1 웹 브라우저를 사용하여 <http://www.cisco.com/go/firepower9300-software> 또는 <http://www.cisco.com/go/firepower4100-software>로 이동합니다.  
FXOS 새시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.
- 단계 2 Cisco.com에서 적절한 펌웨어 패키지를 찾은 후 FXOS 새시에서 액세스할 수 있는 서버로 다운로드합니다.
- 단계 3 FXOS 새시에서 펌웨어 모드로 들어갑니다.  
`fxos-chassis # scopefirmware`
- 단계 4 FXOS 펌웨어 이미지를 FXOS 새시로 다운로드합니다.  
`fxos-chassis /firmware # download image URL`

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

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

- 단계 5 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.



```
fxos-chassis /firmware # show download task image_name detail
```

단계 6 다운로드가 완료되면 다음 명령을 입력하여 펌웨어 패키지의 내용을 볼 수 있습니다.

```
fxos-chassis /firmware # show package image_name expand
```

단계 7 다음 명령을 입력하여 펌웨어 패키지의 버전 번호를 볼 수 있습니다.

```
fxos-chassis /firmware # show package
```

이 버전 번호는 펌웨어 패키지를 설치할 때 다음 단계에서 사용됩니다.

단계 8 펌웨어 패키지를 설치하려면 다음과 같이 합니다.

a) 펌웨어 설치 모드로 들어갑니다.

```
fxos-chassis /firmware # scope firmware-install
```

b) 펌웨어 패키지를 설치합니다.

```
fxos-chassis /firmware/firmware-install # install firmware pack-version version_number
```

시스템에서 펌웨어 패키지를 확인하며, 확인 프로세스를 완료하는 데에는 몇 분 정도 소요될 수 있습니다.

c) **yes**를 입력하여 확인을 계속 진행합니다.

펌웨어 패키지를 확인한 후 시스템에서는 설치 프로세스를 완료하는 데 몇 분 정도 소요될 수 있으며 업데이트 프로세스 중에 시스템이 리부팅된다는 것을 알려줍니다.

d) **yes**를 입력하여 설치를 계속 진행합니다. 업그레이드 프로세스 중에는 FXOS 새시의 전원을 켜다 가 켜지 마십시오.

단계 9 업그레이드 프로세스를 모니터링하려면 다음과 같이 합니다.

```
fxos-chassis /firmware/firmware-install # show detail
```

단계 10 설치가 완료되면 다음 명령을 입력하여 현재 펌웨어 버전을 볼 수 있습니다.

```
fxos-chassis /firmware/firmware-install # top
```

```
fxos-chassis # scope chassis 1
```

```
fxos-chassis /firmware # show sup version
```

다음 예에서는 펌웨어 버전을 1.0.10으로 업그레이드합니다.

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

fxos-chassis /firmware # show download-task fxos-k9-fpr9k-firmware.1.0.10.SPA detail

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

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

```

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

fxos-chassis /firmware # show package

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

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

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

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

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

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

Upgrading FXOS SUP firmware software package version 1.0.10

command executed

```



## 보안 인증 컴플라이언스

- 보안 인증 컴플라이언스, 57 페이지
- FIPS 모드 활성화, 58 페이지
- Common Criteria 모드 활성화, 59 페이지
- SSH 호스트 키 생성, 59 페이지
- IPSec 보안 채널 구성, 60 페이지
- 트러스트 포인트에 대한 정적 CRL 구성, 65 페이지
- 인증서 해지 목록 확인 정보, 66 페이지
- CRL 주기적 다운로드 구성, 70 페이지
- NTP 서버 인증 활성화, 71 페이지
- LDAP 키 링 인증서 설정, 72 페이지
- IP 액세스 목록 구성, 73 페이지
- 클라이언트 인증서 인증 활성화, 74 페이지

## 보안 인증 컴플라이언스

미국 연방 정부 기관은 미 국방성 및 글로벌 인증 기관에서 마련한 보안 표준을 준수하는 장비 및 소프트웨어만 사용해야 할 경우가 있습니다. FXOS 새시는 이러한 보안 인증 표준의 컴플라이언스를 지원합니다.

이러한 표준의 컴플라이언스를 지원하는 기능을 활성화하는 단계는 다음 항목을 참조하십시오.

- FIPS 모드 활성화, 58 페이지
- Common Criteria 모드 활성화, 59 페이지
- IPSec 보안 채널 구성, 60 페이지
- 트러스트 포인트에 대한 정적 CRL 구성, 65 페이지

- CRL 주기적 다운로드 구성, 70 페이지
- NTP 서버 인증 활성화, 71 페이지
- LDAP 키 링 인증서 설정, 72 페이지
- IP 액세스 목록 구성, 73 페이지
- 클라이언트 인증서 인증 활성화, 74 페이지
- 최소 비밀번호 길이 확인 구성, 39 페이지
- 최대 로그인 시도 횟수 설정, 38 페이지
- 사용자 역할, 33 페이지



참고 이러한 항목은 FXOS 새시에서 인증 컴플라이언스를 활성화하는 방법에 대해서만 설명합니다. FXOS 새시에서 인증 컴플라이언스를 활성화한다고 해서 연결된 논리적 디바이스로 컴플라이언스가 자동으로 전파되지는 않습니다.

## FIPS 모드 활성화

FXOS 새시에서 FIPS 모드를 활성화하려면 다음 단계를 수행하십시오.

절차

단계 1 FXOS CLI에서 보안 모드로 들어갑니다.

**scopesystem**

**scopesecurity**

단계 2 FIPS 모드를 활성화합니다.

**enablefips-mode**

단계 3 컨피그레이션을 커밋합니다.

**commit-buffer**

단계 4 시스템을 재부팅합니다.

**connectlocal-mgmt**

**reboot**

다음에 할 작업

FXOS 릴리스 2.0.1 이전에는, 디바이스의 최초 설정 중 생성된 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 Common Criteria 인증 요구 사항을 충족하려면 이러한 과거의 호스트 키를 삭제하고 **SSH 호스트 키 생성, 59 페이지**에 설명된 절차를 사용하여 새 호스트 키를 생성해야 합니다.

이 추가 단계를 수행하지 않으면, FIPS 모드가 활성화되어 디바이스가 리부팅된 후 SSH를 사용하여 Supervisor에 연결할 수 없습니다. FXOS 2.0.1 이상을 사용하여 초기 설정을 수행한 경우 새 호스트 키를 생성할 필요가 없습니다.

## Common Criteria 모드 활성화

FXOS 새시에서 Common Criteria 모드를 활성화하려면 다음 단계를 수행하십시오.

### 절차

- 
- 단계 1 Common Criteria에 대한 **Enable(활성화)** 확인란을 선택합니다.
  - 단계 2 FXOS CLI에서 보안 모드로 들어갑니다.  
**scopesystem**  
**scopesecurity**
  - 단계 3 Common Criteria 모드로 들어갑니다.  
**enablecc-mode**
  - 단계 4 컨피그레이션을 커밋합니다.  
**commit-buffer**
  - 단계 5 시스템을 재부팅합니다.  
**connectlocal-mgmt**  
**reboot**
- 

### 다음에 할 작업

FXOS 릴리스 2.0.1 이전에는, 디바이스의 최초 설정 중 생성된 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 Common Criteria 인증 요구 사항을 충족하려면 이러한 과거의 호스트 키를 삭제하고 [SSH 호스트 키 생성, 59 페이지](#)에 설명된 절차를 사용하여 새 호스트 키를 생성해야 합니다. 이 추가 단계를 수행하지 않으면, Common Criteria 모드가 활성화되어 디바이스가 리부팅된 후 SSH를 사용하여 Supervisor에 연결할 수 없습니다. FXOS 2.0.1 이상을 사용하여 초기 설정을 수행한 경우 새 호스트 키를 생성할 필요가 없습니다.

## SSH 호스트 키 생성

FXOS 릴리스 2.0.1 이전에는, 디바이스의 초기 설정 중 생성된 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 Common Criteria 인증을 준수하려면 이러한 과거의 호스트 키를 삭제하고 새 호스트 키를 생성해야 합니다. 자세한 내용은 [FIPS 모드 활성화, 58 페이지](#) 또는 [Common Criteria 모드 활성화, 59 페이지](#)를 참고하십시오.

과거의 SSH 호스트 키를 삭제하고 인증을 준수하는 새 호스트 키를 생성하려면 다음 단계를 수행하십시오.

## 절차

- 
- 단계 1 FXOS CLI에서 서비스 모드로 들어갑니다.  
**scopesystem**  
**scopeservices**
- 단계 2 SSH 호스트 키를 삭제합니다.  
**deletessh-serverhost-key**
- 단계 3 컨피그레이션을 커밋합니다.  
**commit-buffer**
- 단계 4 SSH 호스트 키 크기를 2048비트로 설정합니다.  
**setssh-serverhost-keyrsa 2048**
- 단계 5 컨피그레이션을 커밋합니다.  
**commit-buffer**
- 단계 6 새 SSH 호스트 키를 생성합니다.  
**createssh-serverhost-key**  
**commit-buffer**
- 단계 7 새 호스트 키 크기를 확인합니다.  
**showssh-serverhost-key**  
호스트 키 크기: 2048
- 

## IPSec 보안 채널 구성

공용 네트워크를 통과하는 데이터 패킷에 대해 엔드 투 엔드 암호화 및 인증 서비스를 제공하기 위해 FXOS 새시에서 IPSec를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 [보안 인증 컴플라이언스, 57 페이지](#)를 참고하십시오.



참고 IKE 및 SA 연결 간에 암호화 키 강도 매칭의 적용을 구성하도록 선택한 경우(아래의 절차에서 sa-strength-enforcement를 yes로 설정):

SA 적용이 활성화된 경우	IKE 협상 키 크기가 ESP 협상 키 크기보다 작은 경우 연결이 실패합니다. IKE 협상 키 크기가 ESP 협상 키 크기보다 크거나 같은 경우 SA 적용 확인이 통과하고 연결이 성공합니다.
SA 적용이 비활성화된 경우	SA 적용 확인이 통과하고 연결이 성공합니다.

IPSec 보안 채널을 구성하려면 다음 단계를 수행하십시오.

절차

단계 1 FXOS CLI에서 보안 모드로 들어갑니다.

**scopesystem**

**scopesecurity**

단계 2 키 링을 생성합니다.

**enterkeyringssp**

**!createcertreqsubject-name<subject-name>ipip**

단계 3 연결된 인증서 요청 정보를 입력합니다.

**entercertreq**

단계 4 국가를 설정합니다.

**setcountrycountry**

단계 5 DNS를 설정합니다.

**setdnsdns**

단계 6 이메일을 설정합니다.

**sete-mailemail**

단계 7 IP 정보를 설정합니다.

**setfi-a-ipfi-a-ip**

**setfi-a-ipv6fi-a-ipv6**

**setfi-b-ipfi-b-ip**

**setfi-b-ipv6fi-b-ipv6**

**setipv6ipv6**

단계 8 지역 정보를 설정합니다.

**setlocality***locality*

단계 9 조직 이름을 설정합니다.

**setorg-name***org-name*

단계 10 조직 단위 이름을 설정합니다.

**setorg-unit-name***org-unit-name*

단계 11 비밀번호를 설정합니다.

**!setpassword**

단계 12 상태를 설정합니다.

**setstate***state*

단계 13 certreq의 주체 이름을 설정합니다.

**setsubject-name***subject-name*

단계 14 종료합니다.

**exit**

단계 15 모듈러스를 설정합니다.

**setmodulus***modulus*

단계 16 인증서 요청의 재생성을 설정합니다.

**setregenerate**{ *yes* | *no* }

단계 17 트러스트 포인트를 설정합니다.

**settrustpoint***interca*

단계 18 종료합니다.

**exit**

단계 19 새로 만든 트러스트 포인트를 입력합니다.

**entertrustpoint***interca*

단계 20 인증서 서명 요청을 생성합니다.

**setcertchain**

예제:

```
-----BEGIN CERTIFICATE-----
MIIF3TCCA8WgAwIBAgIBADANBgkqhkiG9w0BAQsFADBwMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExDDAKBgNVBAAcMA1NKQzEOMAwGA1UECgwFQ2lzY28xDTALBgNV
BAsMBFNuQlUxZzAJBgNVBAMMAkNBMR0wGAYJKoZIhvcNAQkBFgtzc3Bac3NwLm5l
dDAeFw0xNjEyMDgxOTMzNTJaFw0yNjEyMDYxOTMzNTJaMHAcCzAJBgNVBAYTAiVT
MQswCQYDVQQIDAJDQTEMMAoGA1UEBwwDU0pDMQ4wDAYDVQQKDAVDaXNjbzENMA5G
A1UECwwEU1RCVTElMAkGA1UEAwwCQ0ExGjAYBgkqhkiG9w0BCQEWc3Nzc3BzAu
bmV0MIICijANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA2ukWyMLQuLqTvhq7
zFb3Oz/iyDG/ui6mrLIYn8wE3E39XcXA1/x9IHCmxFKNJdD7EbsggfOuy0Bj+Y4s
+uZ1VapBXV/JrAie7bNn3ZYrI29yuyOrlqoi9k9gL/oRBzH18BwBwGHBOz3hGrSK
Yc2yhsq9y/6yI3nSuLZm6ybmUKjTa+B4YuhDTz4hl/I9x/J5nbGiab3vLDKss1nO
xP9+1+Lc690V18/mNPWdjCjDI+U/L9keYs/rbZdRSeXy9kMae42+4FIRHDJjPcSN
Yw1g/gcR2F7QUKRygKckJKXDX2QliGYScTlSHj18O87o5s/pmQAWWRGkKpfDv3oH
cMPgl2T9rC0D8NNcgPXj9PFKfexoGNGwNTO85fK3kjgMODWbdeMG3EihxEEOUPD0
```



```
Fdu0HrTM5lvwb+vr5wE9HsAiMJ8UuujmHqH5mlwyy3Me+cEDHo0hLeNs+AFrqEXQ
e9S+KZC/dq/9zOLpRsVqSfJsAuVl/QdPDbWShjflE/fP2Wj01PqXywQydznmVvgE
wEzAoFg+mIGJm0+q4RDvnpzEviOYNSAGmOkILh5HQ/eYDcxvd0qbORWb31H32ySl
lla6UTT9+vnND1f838fxvNvr8nyGD2S/LVaxnZIO4jcSivtdizbbT8u5B4VcLKIC
x0vkqjo6RvNZJ52sUaD9C3UodTUCAwEAAaOBgTB/MC8GA1UdHwQoMCYwJKAIoCCG
Hmh0dHA6Ly8xOTluMTY4LjQuMjkvcvm9vdGNhLmNybDADBgNVHQ4EFgQU7Jg01A74
jpx8U0APk76pVfYQQ5AwHwYDVR0jBBgwFoAU7Jg01A74jpx8U0APk76pVfYQQ5Aw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAvI8ky2jiXc4wPiMuxIfY
W7DRmszPUWQ7edor7yxuCqzHLVFFOwYRudsyXbv7INR3rJ/X1cRQj9+KidWVWxpo
pFahRhzyxVZ10DhKlzGTQS3jiHgrF3Z8ohWbL15L7PEDlrxMBoJvabPeQRgTmY/n
XZJ7qRYbyop03gUMCaCZ12raJc3/DlpBQ29yweCbUkc9qiHKA0IbnvAxoroHWmBld
94LrJCggfMQTuNJQszJiVVsYJfZ+utlDp2QwfdDv7B0JkwTBjdWRSfotEbc5R18n
BNXYHqxuoNMmqbS3KjCLXcH6xIN8t+Ukfp89hvJt/fluj+s/VJSVZWK4tAWvR7wl
QngCKRJW6FYpzeyNBctiJ07wO+Wt4e3KhJjJDYvA9hFixWcVGDf2r6QW5BYbgGOK
DkHb/gdr/bcdLBKN/PtSJ+prSrpBSaA6rJX8D9UmfhqN/3f+sS1fM4qWORJc6G2
gAeg7AjEQ/0do512vAI8p8idOg/Wv1O17mavzLpcue05cwMCX9fKxkZZ/+7Pk19Y
ZrXS6uMn/CGnViptn0w+uJ1IRj1oulk+/ZyPtBvFHUkFRnhoWj5SMFyds2iaatyI
47N2ViaZBxhU3GICaH+3O+8rs9Kkz9tBZDSnEJVZA6yxaNCVP1bRUO20G3oRTmSx
8iLbjN+BXggxMmG8ssHisgw=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFqDCCA5CgAwIBAgIBBDANBgkqhkiG9w0BAQsFAADwMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExDDAKBgNVBAMcMA1NKQZzEOMAwGA1UECgwFQ2lzY28xDTALBgNV
BAsMBFNUQUIUxCzAJBgNVBAMcMA1NBMR0wGAYJKoZIhvcNAQkBFgtz3BAc3NwLm5l
dDAeFw0xNjEyMTUyMTM0NTRaFw0yNjEyMTUyMTM0NTRaMHwxZzAJBgNVBAYTAiVT
MQswCQYDVQQIDAJDQTEPMA0GA1UECgwGbmV3c3RnMRAwDgYDVVQLDAduZXdzdGJl
MRMwEQYDVQDDAppbnRlcm0xLWNhMSgwJgYJKoZIhvcNAQkBFhlpbnRlcm0xLWNh
QGluGUYybTEtY2EubmV0MIICljANBgkqhkiG9w0BAQEFAAOCAg8AMIICGKCAgEA
wLpNnyEx514P8uDoWKF3IZseghLANsodxuAUmhmwKekd0OpZZxHMw1wSO4IBX5
4itJS0xyXFzPmeptG3OXvNqCcsT+4BXI3DoGgPMULccc4NesHeg2z8+q3SPA6uZh
iseWNVkfnUjixbQEBtrWBiSKnZuOz1cpuBn34gtgeFFoCEXN+EZVpPESiancDVh
8pCPIpc/08ZJ3o9Gw2j0eHJN84sguEDL812ROejQvpmfGQUq11stkIuh+wB+V
VRhUBVG7pV57I6DHeeRp6cDMLXaM3iMTelhdShyo5YUaRJMak/t8kCqhtGXfuLII
E2AkxKXeeveR9n6cpQd5JiNzCT/t9IQL/T/CCqMICRXLFP LCS9o5S5O2B6QFgcTZ
yKR6hsmwe22wpK8QI7/5oWNXl0lb96hHJ7RPbG7RXYqmcLiXY/d2j9/RuNoPJawI
hLkfholdPA28xlnfBlazCmMmdPcBO6cbUQfCj5hSmk3StVQKgjCjaujz55TGd1
GjnxDMX9twzw7Ee51895Xmtr24qqaCXJoW/dPhcIIXRdJPMsTJ4yPG0BieuRwd0p
i8w/rFwbHzv4C9Fthw1JrRxH1yeHJHrLIZgJ5txSaVUIgrgVCJaf6/jrRRWoRjWt
AzvzYq12dZPCcEAYgP7JcaQpvdpuDgq++NgBtygiqECAwEAAANBMD8wDAYDVR0T
BAUwAwEB/zAvBgNVHR8EKDAmMCSglqAghh5odHRwOi8vMTkyLjE2OC40LjI5L2lu
dGVybzS5jcmwwDQYJKoZIhvcNAQELBQADggIBAG/XujJh5G5UWo+cwTSitAezWbJA
h1dAiXZ/OYWZSxkFRliErKdupLqL0ThjnX/wRFfEXbrBQwm5kWAUUDr97D1Uz+2A
8LC5I8SWKXmyf0jUtsnEQbDZb33oVL7yXJk/A0SF0jihpPheMA+YRazalT9xj9KH
PE7nHCJMbb2ptrHUyvbRkSYrSeEqOpQU2+otnFyV3rS9aelgVjuaWyaWoc3lZ1Oi
CC2tJvY3NnM56j5iesxUCeY/SZ2/ECXN7RRBViLHmA3gFKmWf3xeNiKkxmJCxOaa
UWPC1x2V66I8DG9uUzlWyd79O2dy52aAphAHC6hqlzb6v+gw1Tld7UxaqVd8CD5W
ATjNs+ifkJS1h5ERxHjgeurZXOPr+NwPwF+UDzbMXxx+KAAXCI6htCd8Pb3wOUC3
PKvwEXaIcCcxGx71eRLpWPZFyEoi4N2NGE9OXRjz0K/KERZgNhsIW3bQMjcw3aX6
OXskEuKgsayctnWyxVqNnqvuz06kqyubh4+ZgGKZ5LNEXYmGNz3oED1rUN636Tw
SjGAPHgeROzyTFDixCeiaROIGdP/Hwvb0/+uThIe89g8WZ0djTKFUM8uBO3f+II
/cbuyBO1+JrDMq8NkAjkIjlp1c3Wbfcue/qewtcfUBYZ4i53a56UNF5Efrpy/8
B/+07Me/p2y9Luqa
-----END CERTIFICATE-----
ENDOFBUF
```

단계 21 인증서 서명 요청을 표시합니다.  
**showcertreq**

예제:

```
Firepower-chassis# /security/keyring # show certreq
Certificate request subject name: SSP
Certificate request ip address: 192.168.0.111
Certificate request FI A ip address: 0.0.0.0
Certificate request FI B ip address: 0.0.0.0
Certificate request e-mail name:
Certificate request ipv6 address: ::
Certificate request FI A ipv6 address: ::
Certificate request FI B ipv6 address: ::
Certificate request country name: US
State, province or county (full name): CA
Locality name (eg, city): SJC
Organisation name (eg, company): Cisco
Organisational Unit Name (eg, section): Sec
DNS name (subject alternative name):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIICwTCCAakCAQAwVTELMakGA1UEBhMCMVVMxMzA1BjBjNVBAgMAkNBMQwwCgYDVQQH
DANTSkMxMjA1BjBjNVBAgMAkNBMQwwCgYDVQQLEDA1BjBjNVBAgMAkNBMQwwCgYDV
U1AwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDq292Rq3t0laoxPbfE
p/ITKr6rxFhPqSSbtm6sXer//VZFiDTWODockDItuf4Kja215mIS0RyvEYVeRgAs
wbN459wm0BASd8xCjIhsuHDV7yHu539BnvRW6Q2o+gHeSRwckqjClK/tsIxsPkV0
6OduZYXk2bnsLWs6tNk3uzOIT2Q0FcZ1ET66C8fyyKWTrmvcZjDjkMm2nDFsPIX9
39TYPItDkJE3PocqyaCqmT4uobOuvQeLJh/efkBvwhb4BF8vwzRpHWTdjjU5YnR1
qiR4q7j1RmzVFxCDY3IVP/KDBoa5NyCLEUZCEP5QCQFDzIRETZwVOKtxUVG0Nljd
K5TxAgMBAAGgJzA1BjBjNVBAgMAkNBMQwwCgYDVQQNAuCA1NTUicEwKgA
rjANBgkqhkiG9w0BAQsFAAOCAQEArTBoInxXkBYNIVeEoFCqKttu3+Hc7UdyoRM
2L2pjx5OHbQICC+8NRVRMYujTnp67BWuUZZI03dGP4/lbN6bC9P3CvkZdKUsJkN0
m1Ye9dgz7MO/KEcosarmoMl9WB8LlweVdt6ycSdJzs9shOxwT6TAZPwL7gq/1ShF
RJh6sq5W9p6E0SjYefK62E7MatRjDjS8DXoxj6gfn9DqK15iVpkK2QqT5meSGj+
R+20TcUnT0h/S5K/bySEM/3U1gFxQCOzbpPuHkj28kXAVczmTxXEKJBFLVduWN06
DT3u0xImiPR1sqWljpMwbhC+ZGDvtgKjKHToagup9+8R9IMcBQ==
-----END CERTIFICATE REQUEST-----
```

단계 22 IPsec 모드로 들어갑니다.  
**scopeipsec**

단계 23 로그 자세한 정보 레벨을 설정합니다.  
**setlog-levellog\_level**

단계 24 IPsec 연결을 만들고 입력합니다.  
**enterconnectionconnection\_name**

단계 25 IPsec 모드를 tunnel 또는 transport로 설정합니다.  
**setmodetunnel\_or\_transport**

단계 26 로컬 IP 주소를 설정합니다.  
**setlocal-addrrip\_address**

단계 27 원격 IP 주소를 설정합니다.  
**setremote-addrrip\_address**

단계 28 터널 모드를 사용하는 경우 원격 서버넷을 설정합니다.

**setremote-subnetip/mask**

단계 29 (선택 사항) 원격 ID를 설정합니다.

**setremote-ike-ident***remote\_identity\_name*

단계 30 키 링 이름을 설정합니다.

**setkeyring-name***name*

단계 31 (선택 사항) 키 링 비밀번호를 설정합니다.

**setkeyring-passwd***passphrase*

단계 32 (선택 사항) IKE-SA 수명을 분 단위로 설정합니다.

**setike-rekey-time***minutes*

*minutes* 값은 60~1440의 정수일 수 있습니다.

단계 33 (선택 사항) Child SA 수명을 분 단위로 설정합니다(30-480).

**setesp-rekey-time***minutes*

*minutes* 값은 30~48의 정수일 수 있습니다.

단계 34 (선택 사항) 초기 연결 중에 수행할 재전송 시퀀스의 수를 설정합니다.

**setkeyringtries***retry\_number*

*retry\_number* 값은 1~5의 정수일 수 있습니다.

단계 35 (선택 사항) 인증서 해지 목록 확인을 활성화 또는 비활성화합니다.

**setrevoke-policy**{ *relaxed* | *strict* }

단계 36 연결을 활성화합니다.

**setadmin-state***enable*

단계 37 모든 연결을 다시 로드합니다.

**reload-conns**

단계 38 (선택 사항) 기존 트러스트 포인트 이름을 IPsec에 추가합니다.

**createauthority***trustpoint\_name*

단계 39 IKE 및 SA 연결 간 암호화 키 강도 매칭의 적용을 구성합니다.

**setsa-strength-enforcement***yes\_or\_no*

## 트러스트 포인트에 대한 정적 CRL 구성

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

CRL(Certification Revocation List) 정보를 사용하여 피어 인증서를 검증하도록 FXOS 새시를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 [보안 인증 컴플라이언스, 57 페이지](#)를 참고하십시오.

CRL 정보를 사용하여 피어 인증서를 검증하려면 다음 단계를 수행하십시오.

절차

- 단계 1 FXOS CLI에서 보안 모드로 들어갑니다.  
**scopesecurity**
- 단계 2 트러스트 포인트 모드로 들어갑니다.  
**scopetrustpointtrustname**
- 단계 3 해지 모드로 들어갑니다.  
**scoperevoke**
- 단계 4 CRL 파일을 다운로드합니다.  
**importcrlprotocol://user\_id@CA\_or\_CRL\_issuer\_IP/tmp/DoDCA1CRL1.crl**
- 단계 5 (선택 사항) CRL 정보 가져오기 프로세스의 상태를 표시합니다.  
**showimport-taskdetail**
- 단계 6 인증서 해지 메서드를 CRL-only로 설정합니다.  
**setcertrevokemethod{crl}**

## 인증서 해지 목록 확인 정보

IPSec 및 안전한 LDAP 연결에서 CRL(Certificate Revocation List) 확인 모드를 Strict 또는 Relaxed로 구성할 수 있습니다.

동적(정적이 아님) CRL 정보는 X.509 인증서의 CDP 정보에서 수집되며 동적 CRL 정보를 나타냅니다. 정적 CRL 정보는 시스템 관리에 의해 수동으로 다운로드되며, FXOS 시스템에서 로컬 CRL 정보를 나타냅니다. 동적 CRL 정보는 인증서 체인에서 현재 처리 중인 인증서에 대해서만 처리됩니다. 정적 CRL은 전체 피어 인증서 체인에 적용됩니다.

안전한 LDAP 및 IPSec 연결을 위한 인증서 해지 확인을 활성화 또는 비활성화하는 단계는 [IPSec 보안 채널 구성, 60 페이지](#) 및 [LDAP 제공자 생성, 130 페이지](#) 섹션을 참조하십시오.

다음 표에서는 인증서 해지 목록 확인 설정 및 인증서 검증에 따라 연결 결과를 설명합니다.

표 4: 로컬 정적 **CRL** 없이 정적으로 설정된 인증서 해제 확인 모드

로컬 정적 <b>CRL</b> 없음	LDAP 연결	IPSec 연결
피어 인증서 체인 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인에서 CDP 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음

로컬 정적 CRL 없음	LDAP 연결	IPSec 연결
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 성공
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	syslog 메시지와 함께 연결 실패	연결 성공
피어 인증서 체인의 CDP 중 다운로드할 수 없는 것이 있음	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 성공
인증서에 CDP가 있지만 CDP 서버가 다운됨	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 성공
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 성공

표 5: 로컬 정적 CRL과 함께 Strict로 설정된 인증서 해지 확인 모드

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
피어 인증서 체인 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인에서 CDP 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패

로컬 정적 <b>CRL</b> 있음	<b>LDAP</b> 연결	<b>IPSec</b> 연결
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨	연결 성공	연결 성공
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공
피어 인증서 체인의 CDP 중 다운로드할 수 없는 것이 있음	연결 성공	연결 성공
인증서에 CDP가 있지만 CDP 서버가 다운됨	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음	연결 성공	연결 성공

표 6: 로컬 정적 **CRL** 없이 **Relaxed**로 설정된 인증서 해제 확인 모드

로컬 정적 <b>CRL</b> 없음	<b>LDAP</b> 연결	<b>IPSec</b> 연결
피어 인증서 체인 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인에서 CDP 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨	연결 성공	연결 성공

로컬 정적 CRL 없음	LDAP 연결	IPSec 연결
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공
피어 인증서 체인의 CDP 중 다 운로드할 수 없는 것이 있음	연결 성공	연결 성공
인증서에 CDP가 있지만 CDP 서 버가 다운됨	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가 동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있 음	연결 성공	연결 성공

표 7: 로컬 정적 CRL과 함께 **Relaxed**로 설정된 인증서 해제 확인 모드

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
피어 인증서 체인 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인에서 CDP 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인의 루트 CA 인 증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인 증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락됨	연결 성공	연결 성공
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공
피어 인증서 체인의 CDP 중 다 운로드할 수 없는 것이 있음	연결 성공	연결 성공

로컬 정적 CRL 있음	LDAP 연결	IPSec 연결
인증서에 CDP가 있지만 CDP 서버가 다운됨	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음	연결 성공	연결 성공

## CRL 주기적 다운로드 구성

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

이 기능과 함께 다음 프로토콜 및 인터페이스를 사용할 수 있습니다.

- FTP
- SCP
- SFTP
- TFTP
- USB



참고

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



참고

기간은 1시간 간격으로만 구성할 수 있습니다.

CRL 주기적 다운로드를 구성하려면 다음 단계를 수행하십시오.

시작하기 전에

CRL 정보를 사용하여 피어 인증서를 검증하도록 FXOS 새시를 이미 구성했는지 확인하십시오. 자세한 내용은 [트러스트 포인트에 대한 정적 CRL 구성](#), 65 페이지를 참고하십시오.



## 절차

단계 1 FXOS CLI에서 보안 모드로 들어갑니다.

**scopesecurity**

단계 2 트러스트 포인트 모드로 들어갑니다.

**scopetrustpoint**

단계 3 해지 모드로 들어갑니다.

**scoperevoke**

단계 4 해지 컨피그레이션을 수정합니다.

**shconfig**

단계 5 원하는 컨피그레이션을 설정합니다.

예제:

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

단계 6 컨피그레이션 파일을 종료합니다.

**exit**

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

예제:

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

Import task:

File Name	Protocol	Server	Port	Userid	State
rootCA.crl	Scp	182.23.33.113	0	myname	Downloading

## NTP 서버 인증 활성화

FXOS 새시에서 NTP 서버 인증을 활성화하려면 다음 단계를 수행하십시오.



참고

- NTP 인증 기능은 활성화될 경우 모든 구성된 서버에 대해 전역적으로 적용됩니다.
- NTP 서버 인증에는 SHA1만 지원됩니다.
- 서버를 인증하려면 키 ID 및 키 값이 필요합니다. MD(message digest)를 계산할 때 어떤 키 값을 사용할지를 클라이언트 및 서버에 알려줄 때 키 ID가 사용됩니다. 이 키 값은 `ntp-keygen`을 사용하여 파생된 고정값입니다.

### 절차

- 단계 1 `ntp 4.2.8p8`을 다운로드합니다.
- 단계 2 `ntpd openssl`을 활성화하여 NTP 서버를 설치합니다.
- 단계 3 NTP 키 IDs 및 키 값을 생성합니다.  
**`ntp-keygen-M`**  
생성된 키를 다음 단계에 사용합니다.
- 단계 4 FXOS CLI에서 NTP 서버를 생성합니다.  
**`creatntp-serverserver_id`**
- 단계 5 NTP 서버를 입력합니다.  
**`scopentp-serverserver_id`**
- 단계 6 SHA1 Key ID를 설정합니다.  
**`setntp-sha1-key-idkey_id`**
- 단계 7 SHA1 Key String을 설정합니다.  
**`setntp-sha1-key-stringkey_string`**
- 단계 8 NTP 인증을 활성화합니다.  
**`enablenntp-authentication`**

## LDAP 키 링 인증서 설정

FXOS 새시에서 TLS 연결을 지원하기 위해 안전한 LDAP 클라이언트 키 링 인증서를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 [보안 인증 컴플라이언스, 57 페이지](#)를 참고하십시오.



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

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

안전한 LDAP 연결(SSL 활성화)을 위해 LDAP 서버 정보는 CC 모드에서 DNS 정보여야 합니다.

안전한 LDAP 클라이언트 키 링 인증서를 구성하려면 다음 단계를 수행하십시오.

#### 절차

**단계 1** FXOS CLI에서 보안 모드로 들어갑니다.

**scopesecurity**

**단계 2** LDAP 모드로 들어갑니다.

**scopeldap**

**단계 3** LDAP 서버 모드로 들어갑니다.

**enterserver**{server\_ip|server\_dns}

**단계 4** LDAP 키 링을 설정합니다.

**setkeyring**keyring\_name

**단계 5** 컨피그레이션을 커밋합니다.

**commit-buffer**

## IP 액세스 목록 구성

기본적으로 FXOS 새시는 로컬 웹 서버에 대한 모든 액세스를 거부합니다. 각 IP 블록에 대해 허용된 서비스 목록으로 IP 액세스 목록을 구성해야 합니다.

IP 액세스 목록은 다음 프로토콜을 지원합니다.

- HTTPS
- SNMP
- SSH

IP 주소(v4 또는 v6) 각 블록에서 각 디바이스에 대해 최대 25개의 서로 다른 서브넷을 구성할 수 있습니다. 서브넷 0과 접두사 0은 서비스에 대한 무제한 액세스를 허용합니다.

## 절차

단계 1 FXOS CLI에서 서비스 모드로 들어갑니다.

```
scopesystem
```

```
scopesecurity
```

단계 2 액세스를 활성화할 서비스에 대한 IP 블록을 생성합니다.

IPv4의 경우:

```
createip-blockipprefix [0-32]https/snmp/ssh
```

IPv6의 경우:

```
createipv6-blockipprefix [0-28]https/snmp/ssh
```

## IPv4:

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

## IPv6:

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

## 클라이언트 인증서 인증 활성화

LDAP와 함께 클라이언트 인증서를 사용하여 사용자의 HTTPS 액세스를 인증하도록 시스템을 설정할 수 있습니다. FXOS 새시의 기본 인증 컨피그레이션은 자격 증명 기반입니다.



**참고** 인증서 인증이 활성화된 경우, 이것이 HTTPS에 대해 허용되는 유일한 인증 형식입니다. 클라이언트 인증서 인증 기능의 FXOS 2.1.1 릴리스에서는 인증서 해지 확인이 지원되지 않습니다.

이 기능을 사용하려면 클라이언트 인증서에서 다음 요구 사항을 충족해야 합니다.

- X509 특성 Subject Alternative Name - Email(주체 대체 이름 - 이메일)에 사용자 이름을 포함해야 합니다.
- Supervisor의 트러스트 포인트로 인증서를 가져온 루트 CA가 클라이언트 인증서에 서명해야 합니다.

### 절차

**단계 1** FXOS CLI에서 서비스 모드로 들어갑니다.

**scopesystem**

**scopesecurity**

**단계 2** (선택 사항) HTTPS 인증에 대한 옵션을 확인합니다.

**sethttpsauth-type**

예제:

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

**단계 3** HTTPS 인증을 클라이언트 기반으로 설정합니다.

**sethttpsauth-typecert-auth**

**단계 4** 컨피그레이션을 커밋합니다.

**commit-buffer**





# 8 장

## 시스템 관리

---

- 관리 IP 주소 변경, 77 페이지
- 애플리케이션 관리 IP 변경, 79 페이지
- Pre-Login 배너, 80 페이지
- FXOS 새시 리부팅, 83 페이지
- FXOS 새시 전원 끄기, 83 페이지
- 신뢰할 수 있는 ID 인증서 설치, 84 페이지
- 소프트웨어 장애에서 복구, 89 페이지
- 손상된 파일 시스템에서 복구, 93 페이지

## 관리 IP 주소 변경

시작하기 전에

FXOS 새시의 관리 IP 주소를 FXOS CLI에서 변경할 수 있습니다.



---

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

---

절차

---

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

단계 2 다음과 같이 IPv4 관리 IP 주소를 구성합니다.

- a) 패브릭 인터커넥트 a의 범위를 설정합니다.  
Firepower-chassis# **scope fabric-interconnect a**

- b) 다음 명령을 입력하여 현재 관리 IP 주소를 확인합니다.  
Firepower-chassis /fabric-interconnect # **show**
- c) 다음 명령을 입력하여 새로운 관리 IP 주소 및 게이트웨이를 구성합니다.  
Firepower-chassis /fabric-interconnect # **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 다음과 같이 IPv6 관리 IP 주소를 구성합니다.

- a) 패브릭 인터커넥트 a의 범위를 설정합니다.  
Firepower-chassis# **scope fabric-interconnect a**
- b) 관리 IPv6 컨피그레이션의 범위를 설정합니다.  
Firepower-chassis /fabric-interconnect # **scope ipv6-config**
- c) 다음 명령을 입력하여 현재 관리 IPv6 주소를 확인합니다.  
Firepower-chassis /fabric-interconnect/ipv6-config # **show ipv6-if**
- d) 다음 명령을 입력하여 새로운 관리 IP 주소 및 게이트웨이를 구성합니다.  
Firepower-chassis /fabric-interconnect/ipv6-config # **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 #
```

## 애플리케이션 관리 IP 변경

FXOS 새시에 연결된 애플리케이션의 관리 IP 주소를 FXOS CLI에서 변경할 수 있습니다. 그렇게 하려면 먼저 FXOS 플랫폼 레벨에서 IP 정보를 변경한 다음, 애플리케이션 레벨에서 IP 정보를 변경해야 합니다.



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

### 절차

**단계 1** FXOS CLI에 연결합니다. ([FXOS CLI 액세스](#), 12 페이지를 참조하십시오.)

**단계 2** 논리적 디바이스로 범위를 지정합니다.

**Scope ssa**

**scopelogical-device***asa\_or\_ftd*

**단계 3** 관리 부트스트랩으로 범위를 지정합니다.

독립형 컨피그레이션의 경우:

**scopemgmt-bootstrap***asa\_or\_ftd*

클러스터링된 컨피그레이션의 경우:

**scopecluster-bootstrap**

**단계 4** 새 관리 부트스트랩 매개변수를 구성합니다.

독립형 컨피그레이션의 경우:

**setvirtualipv4\_or\_6ip\_address***masknetwork\_maskgatewaygateway\_ip\_address*

클러스터링된 컨피그레이션의 경우:

**setvirtualipv4\_or\_6ip\_address***poolstart\_ip\_end\_ipmasknetwork\_maskgatewaygateway\_ip\_address*

**단계 5** ssa 모드로 다시 범위를 지정합니다.

**Scope ssa**

**단계 6** slot로 범위를 지정합니다.

**scopeslots***slot\_number*

**단계 7** 애플리케이션 범위를 지정합니다.

**scopeapp-instance***asa\_or\_ftd*

**단계 8** 관리 부트스트랩 정보를 지웁니다.

**clearmgmt-bootstrap**

참고 클러스터링된 컨피그레이션의 경우 FXOS 새시에 연결된 각 애플리케이션에 대해 6~8단계를 반복해야 합니다. 새시 간 클러스터 컨피그레이션의 경우 두 새시의 각 애플리케이션에 대해 이러한 단계를 반복해야 합니다.

단계 9 관리 부트스트랩 컨피그레이션 범위를 종료합니다.

**exit**

단계 10 컨피그레이션을 커밋합니다.

**commit-buffer**

단계 11 보안 모듈의 콘솔에 연결합니다. (애플리케이션 콘솔 또는 데코레이터에 연결, 187 페이지 참조) 클러스터에 대한 관리 IP 정보를 변경하는 경우 마스터 애플리케이션 콘솔에 로그인해야 합니다.

단계 12 가상 IP, 마스크 및 게이트웨이 값을 3단계에 사용된 정확한 값으로 변경합니다.

독립형 컨피그레이션의 경우:

**setvirtualipip\_addressnetmasknetwork\_maskgwgateway\_ip\_address**

클러스터링된 컨피그레이션의 경우:

**setvirtualvip\_addresspoolstart\_ip\_end\_ipmasknetwork\_maskgatewaygateway\_ip\_address**

단계 13 컨피그레이션을 커밋합니다.

**commit-buffer**

## Pre-Login 배너

Pre-login 배너가 있으면 사용자가 Firepower Chassis Manager에 로그인할 때 시스템에 배너 텍스트가 표시됩니다. 사용자가 메시지 화면에서 **OK**(확인)를 클릭하면 사용자 이름과 비밀번호 프롬프트 창이 표시됩니다. Pre-login 배너가 구성되어 있지 않으면 사용자 이름과 비밀번호 프롬프트 창이 바로 표시됩니다.

사용자가 FXOS CLI에 로그인하면, 비밀번호 프롬프트가 나타나기 전에 배너 텍스트(구성한 경우)가 표시됩니다.

## Pre-Login 배너 생성

절차

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

단계 2 보안 모드를 입력합니다.

Firepower-chassis# **scopesecurity**

단계 3 배너 보안 모드로 들어갑니다.

Firepower-chassis /security # **scopebanner**

- 단계 4 다음 명령을 입력하여 pre-login 배너를 만듭니다.  
Firepower-chassis /security/banner # **create pre-login-banner**
- 단계 5 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하기 전에 FXOS에서 사용자에게 표시해야 할 메시지를 지정합니다.  
Firepower-chassis /security/banner/pre-login-banner\* # **set message**  
pre-login 배너 메시지 텍스트를 입력하기 위한 대화 상자가 열립니다.
- 단계 6 프롬프트에서 pre-login 배너 메시지를 입력합니다. 이 필드에는 어떤 표준 ASCII 문자도 사용할 수 있습니다. 여러 줄의 텍스트를 입력할 수 있으며 각 줄의 최대 문자 수는 192자입니다. 줄 사이에 **Enter**를 누릅니다.  
입력 다음 줄에 ENDOFBUF를 입력하고 **Enter**를 눌러 완료합니다.  
메시지 설정 대화 상자를 취소하려면 Ctrl 및 C를 누릅니다.
- 단계 7 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
Firepower-chassis /security/banner/pre-login-banner\* # **commit-buffer**

다음 예에서는 pre-login 배너를 생성합니다.

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

## Pre-Login 배너 수정

### 절차

- 단계 1 FXOS CLI에 연결합니다(FXOS CLI 액세스, 12 페이지 참고).
- 단계 2 보안 모드를 입력합니다.  
Firepower-chassis# **scopesecurity**
- 단계 3 배너 보안 모드로 들어갑니다.  
Firepower-chassis /security # **scopebanner**
- 단계 4 pre-login-banner 배너 보안 모드로 들어갑니다.  
Firepower-chassis /security/banner # **scope pre-login-banner**
- 단계 5 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하기 전에 FXOS에서 사용자에게 표시해야 할 메시지를 지정합니다.  
Firepower-chassis /security/banner/pre-login-banner # **set message**

pre-login 배너 메시지 텍스트를 입력하기 위한 대화 상자가 열립니다.

**단계 6** 프롬프트에서 pre-login 배너 메시지를 입력합니다. 이 필드에는 어떤 표준 ASCII 문자도 사용할 수 있습니다. 여러 줄의 텍스트를 입력할 수 있으며 각 줄의 최대 문자수는 192자입니다. 줄 사이에 **Enter**를 누릅니다.

입력 다음 줄에 ENDOFBUF를 입력하고 **Enter**를 눌러 완료합니다.

메시지 설정 대화 상자를 취소하려면 Ctrl 및 C를 누릅니다.

**단계 7** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

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

다음 예에서는 pre-login 배너를 수정합니다.

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

## Pre-Login 배너 삭제

절차

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

**단계 2** 보안 모드를 입력합니다.

```
Firepower-chassis# scopesecurity
```

**단계 3** 배너 보안 모드로 들어갑니다.

```
Firepower-chassis /security # scopebanner
```

**단계 4** 시스템에서 pre-login 배너를 삭제합니다.

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

**단계 5** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/banner* # commit-buffer
```

다음 예에서는 pre-login 배너를 삭제합니다.

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

## FXOS 새시 리부팅

### 절차

단계 1 새시 모드로 들어갑니다.

**scope chassis 1**

단계 2 다음 명령을 입력하여 새시의 전원을 끕니다.

**reboot [ reason ] [no-prompt]**

참고 **[no-prompt]** 키워드를 사용하면 명령을 입력한 후 새시가 즉시 리부팅됩니다. **[no-prompt]** 키워드를 사용하지 않으면 **commit-buffer** 명령을 입력할 때까지 시스템이 리부팅되지 않습니다.

시스템에 구성된 모든 논리적 디바이스가 정상적으로 셧다운된 후 각 보안 모듈/엔진의 전원이 꺼지고, 마지막으로 FXOS 새시의 전원이 꺼진 후 재시작됩니다. 이 프로세스는 보통 15~20분 정도 걸립니다.

단계 3 리부팅 프로세스를 모니터링하려면:

**scope chassis 1**

**show fsm status**

## FXOS 새시 전원 끄기

### 절차

단계 1 새시 모드로 들어갑니다.

**scope chassis 1**

단계 2 다음 명령을 입력하여 새시의 전원을 끕니다.

**shutdown [ reason ] [no-prompt]**

참고 **[no-prompt]** 키워드를 사용하면 명령을 입력한 후 새시가 즉시 셧다운됩니다. **[no-prompt]** 키워드를 사용하지 않으면 **commit-buffer** 명령을 입력할 때까지 시스템이 셧다운되지 않습니다.

시스템에 구성된 모든 논리적 디바이스가 정상적으로 셧다운된 후 각 보안 모듈/엔진의 전원이 꺼지고, 마지막으로 FXOS 새시의 전원이 꺼집니다. 이 프로세스는 보통 15~20분 정도 걸립니다. 새시가 성공적으로 셧다운되면 새시에서 전원 플러그를 뽑을 수 있습니다.

단계 3 셧다운 프로세스를 모니터링하려면:

**scope chassis 1**

**show fsm status**

## 신뢰할 수 있는 ID 인증서 설치

초기 컨피그레이션 이후 FXOS 새시 웹 애플리케이션에서 사용하기 위한 자체 서명 SSL 인증서가 생성됩니다. 인증서가 자체 서명된 것이므로 클라이언트 브라우저에서 이를 자동으로 신뢰하지 않습니다. 새 클라이언트 브라우저는 FXOS 새시 웹 인터페이스에 처음 액세스할 때, FXOS 새시에 액세스하려면 먼저 인증서를 수락하도록 사용자에게 요구하는 SSL 경고를 표시합니다. FXOS CLI를 사용하여 CSR(Certificate Signing Request)을 생성하고 FXOS 새시에서 사용할 결과 ID 인증서를 설치하려면 다음 절차를 사용할 수 있습니다. 이 ID 인증서를 사용하면 클라이언트 브라우저가 연결을 신뢰하며 경고 없이 웹 인터페이스를 표시합니다.

### 절차

- 
- 단계 1 FXOS CLI에 연결합니다. (FXOS CLI 액세스, 12 페이지를 참조하십시오.)
- 단계 2 보안 모듈을 입력합니다.  
**scopesecurity**
- 단계 3 키 링을 생성합니다.  
**createkeyringkeyring\_name**
- 단계 4 개인 키의 모듈러스 크기를 설정합니다.  
**setmodulussize**
- 단계 5 컨피그레이션을 커밋합니다.  
**commit-buffer**
- 단계 6 CSR 필드를 구성합니다. 기본 옵션(예: subject-name)으로 인증서를 생성할 수도 있고, 인증서에 로케일 및 조직과 같은 정보를 포함하도록 허용하는 좀 더 고급 옵션을 선택적으로 사용할 수도 있습니다. CSR 필드를 구성할 때 인증서 비밀번호를 입력하라는 프롬프트가 표시됩니다.  
**createcertreqcertreqsubject\_name**  
*password*  
**setcountrycountry**  
**setstatestate**  
**setlocalitylocality**  
**setorg-nameorganization\_name**  
**setorg-unit-nameorganization\_unit\_name**  
**setsubject-namesubject\_name**
- 단계 7 컨피그레이션을 커밋합니다.  
**commit-buffer**
- 단계 8 인증 증명에 제공할 CSR을 내보냅니다.  
a) 전체 CSR을 표시합니다.  
**showcertreq**

b) "-----BEGIN CERTIFICATE REQUEST-----"로 시작하고(포함) "-----END CERTIFICATE REQUEST-----"로 끝나는(포함) 출력을 복사합니다.

예제:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC6zCCAdMCAQAwdzELMAkGA1UEBhMCVVMxEzARBgNVBAgMCKNhbg1mb3JuaWEw
ETAPBgNVBACMFNhb3NlMRYwFAYDVKQKDA1DaXNjb3B0eXN0ZW1zMQwwCgYD
VQQLDANUQUxGjAYBgNVBAMMEWZwNDEyMC50ZXN0LmXvY2F5S0ZlbnB1eXN0Zl
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs0ON5gagkfZ2fi4JVEANG+7YGgcHbnUt7LpV
yMChnKOPJjBwkUMNQA1mQsRQDcbJ232/sK0fmsnyqOL8JzC7itxeVEZRyz7/ax7W
GNveg/XP+zd03nt4GXM63FsrPcPmA7EwgqDSLoShtBEV10hhf4+Nw4pKCz+eSSkS
JkTB1ZHaKV9bttYg3kf/UEUUGk/EyrVq3B+u2DsooPVq76mTm8BwYmGhbJv4Pmu
RjWE88yEvVwH7JTEij9OvxbatjDjVSHZBURtCanvyBvGuLP/Q/Nmv3Lo3G9ITbL
L5gIYZVatTxp6HTUezH2MIzOavU6d1tB9rnyxgGth5dPV0dhQIDAQABoC8wLQYJ
KoZIHvCNAQkOMSAwHjAcBgNVHREFTATghFmcDQxMjAudGVzdC5sb2NhbDANBgkq
hkiG9w0BAQsFAAOCAQEAZUfCbwx9vt5aVdCL+tATu5xFE3LA310ck6Gj1Nv6W/6r
jBNLxusYilrZzCW+CgnvNs4ArqYGyNVBySOavJO/VvQ1KfyxxJ10Ikyx3RzEjgK0
zzyoYrG+EZXCS5ShiraS8HuWvE2wFM2wwWNtHWtvcQy55+/hDPPD2Bv8pQOC2Zng3I
kLfG1dxWf1xAxLzf5J+AuIQ0CM5HzM9Zm8zREoWT+xHtLSqAgg/aCuomN9/vEwyU
OYfoJmVaqC6AZyUnMfufCoyuLpLwgkxB0gyaRdnea5RhiGjYQ21DXDYdJExp7rCx9
+6bvD11n70JCegHdCWtP75SaNyaBEPk00365rTckbw==
-----END CERTIFICATE REQUEST-----
```

단계 9 certreq 모드를 종료합니다.  
**exit**

단계 10 키 링 모드를 종료합니다.  
**exit**

단계 11 참고 FXOS로 가져오려면 모든 인증서는 Base64 형식이어야 합니다. 인증 증명에서 수신한 인증서 또는 체인의 형식이 다른 경우, OpenSSL과 같은 SSL 도구로 우선 변환해야 합니다. 인증서 체인을 유지할 새 트러스트 포인트를 생성합니다.  
**createtrustpointtrustpoint\_name**

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

단계 13 참고 중간 인증서를 사용하는 인증 증명의 경우 루트 인증서와 중간 인증서를 결합해야 합니다. 텍스트 파일에서 맨 위에 루트 인증서를 붙여넣고, 그 뒤에 체인의 각 중간 인증서를 붙여넣습니다(모든 BEGIN CERTIFICATE 및 END CERTIFICATE 플래그 포함). 전체 텍스트 블록을 트러스트 포인트에 복사하여 붙여넣습니다. 화면이 지침에 따라, 8단계에서 복사한 CSR 출력을 입력합니다.

예제:

```
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
>-----BEGIN CERTIFICATE-----
>MIICDTCCABOgAwIBAgIQYIutxPDPw6B0p3uKNgJHZDAKBggqhkjOPQQDAjBTMRUw
>EwYKZCImiZPyLGQBGRYFbG9jYXVwGDAWBgOJkiaJk/IsZAEZFghuYWF1c3RpbjEg
>MB4GA1UEAxMXbmfhdXN0aW4tTtkFBVVNUSU4tUEMtQ0EwHhcNMjUwNzI4MTc1NjU0
>WhcNMjUwNzI4MTc1NjU0WjBTMRUwEwYKZCImiZPyLGQBGRYFbG9jYXVwGDAWBgOJ
>kiaJk/IsZAEZFghuYWF1c3RpbjEgMB4GA1UEAxMXbmfhdXN0aW4tTtkFBVVNUSU4t
>UEMtQ0EwHhcNMjUwNzI4MTc1NjU0WjBTMRUwEwYKZCImiZPyLGQBGRYFbG9jYXVw
>GXRpXWIEyuiBM4eQRoqZKncJUKmlxmqlubaDHPJ5TMGfJQYszLBRJPq+mdrKcDl
>o2kwZzATBgrBgEEAYI3FAIEBh4EAEMAQTAOBgNVHQ8BAf8EBAMCAyYwDwYDVR0T
>AQH/BAUwAwEB/zAdBgNVHQ4EFgQUyInbDHPPrFwEEBcbxGSgQW7pOVIkwEAYJKwYB
>BAGCNxUBBAMCAQAwCgYIKoZIzj0EAwIDSAAwRQIhAP++QJTUmniB/AxPDDN63Lqy
>18odMDoFTkG4p3Tb/2yMAiAtMYh1sv1gCxsQV0w0xZVRugSdoOak6n7wCjTFX9jr
>RA==
>-----END CERTIFICATE-----
```

```
>ENDOFBUF
```

단계 14 컨피그레이션을 커밋합니다.

```
commit-buffer
```

단계 15 트러스트 포인트 모드를 종료합니다.

```
exit
```

단계 16 키 링 모드로 들어갑니다.

```
scopekeyringkeyring_name
```

단계 17 13단계에서 생성한 트러스트 포인트를 CSR에 대해 생성한 키 링과 연결합니다.

```
settrustpointtrustpoint_name
```

단계 18 서명한 서버용 ID 인증서를 가져옵니다.

```
setcert
```

단계 19 인증 증명에서 제공한 ID 인증서의 내용을 붙여넣습니다.

예제:

```
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
>-----BEGIN CERTIFICATE-----
>MIIe8DCCBjAgAwIBAgITRQAAAAreh1UWgiTzvgAAAAACjAKBggqhkJOPQQDAjBT
>MRUwEwYKCZImiZPyLQGvBgRYFbG9jYWwxGDAWBoJkiaJk/IsZAEZFghuYWF1c3Rp
>bjEgMB4GA1UEAxMXbFhdXN0aW4tTktFBVVNUSU4tUEMtcQ0EwHhcNMTYwNDI4MTMw
>OTU0WhcNMTgwNDI4MTMwOTU0WjB3MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2F5
>aWZvc2V5TERMA8GA1UEBxMIU2FuIEpvc2UxXjFjAUBGNVBAoTDUNpc2NvIFN5c3Rl
>bXMxDDAKBgNVBAsTA1RBQzEaMBGGA1UEAxMRZnA0MTIwLnRlc3QubG9jYWwwggEi
>MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCzQ43mBqCR9nZ+Lg1UQA0b7tga
>BwdudS3sulXIwKGco48mMHCRCw1ADWZCxFANxsnbfb+wrR8xKfKo4vwnMLuK3F5U
>RlHLPv9rHtYY296D9c/7N3Tee3gzczrcWys9w+YDsTCCoNIuhKG0ERXXSGF/j43D
>ikoJn55JKRlImRMHVkdopXlu21iDeR/9QRRSCT8TKtWrcH67Y0yig9WrvqZObwHBg
>yodskS/g+a5GNyTzzIS9XAfs1MSKP06/Ftq2MONVIkdKFRG0Jqe/IG8a4s/9D82a
>/cujcb0hNsSvmAhh1Vq1PGnodNR7MFYwgjM5q9Tp3W0H2ufLGAA2H109XR2FAGMB
>AAGjggjYMIcVDacBgNVHREEFtATghFmcDQxMjAudGVzdC5sb2NhbDAdBgNVHQ4E
>fGQU/1WpstiEYExs8D1ZWcuHwZwPt5QwHwYDVR0jBBgwFoAUyInbDHPzFwEEBcbx
>GSgQW7pOVIkwgdwGA1UdHwSB1DCB0TCBzqCBy6CByIaBxWxkYXA6Ly8vQ049bmFh
>dXN0aW4tTktFBVVNUSU4tUEMtcQ0EesQ049bmFhdXN0aW4tcGMsQ049Q0RQLENOFVB1
>YmXpYyUyMEtleSUyMFn1cnZpY2VzLENOPVN1cnZpY2VzLENOPUNvbmZpZ3VyYXRp
>b24sREM9bmFhdXN0aW4sREM9bG9jYWw/Y2VydGlmawNhdGVsZXZvY2F0aW9uTG1z
>dD9iYXN1P29iamVjdENsYXNzPWNSTERpc3RyaWJldGlvb1BvaW50MlHMBggrBgEF
>BQcBAQSBvzCBvDCBuQYIKYWBQUHMAKGaxsZGFwOi8vL0NOPW5hYXVzdGluLU5B
>QVVTVE10LVBDLUNBLENOFUFJQSxDTj1QdWJsaWw1MjBlZXk1MjBTZXJ2aWN1cyxD
>Tj1TZXJ2aWN1cyxDTj1Db25maWd1cmF0aW9uLERDPW5hYXVzdGluLERDPWxvY2Fs
>P2NBQ2VydGlmawNhdGU/YmFzZT9vYmplY3RDbGFzcmZ1ZjZlZj0aWZpY2F0aW9uQXV0
>aG9yaXR5MCEGCSsGAQQBgjcUAQQUHhIAVwB1AGIAUwB1AHIAAdgB1AHIdG9YDVR0P
>AQH/BAQDAgWgMBMGA1UdJQOMMAoGCCsGAQUFBwMBMAoGCCqGSM49BAMCA0gAMEUC
>IFew7NcJirEtFRvxyjkQ4/dVo2oI6CRB308WQbYHNUu/AieA7UdObiSJBG/PBzjm
>sgoIK60akbjotUvUdUd9b6K1Uw=
>-----END CERTIFICATE-----
>ENDOFBUF
```

단계 20 키 링 모드를 종료합니다.

```
exit
```

단계 21 보안 모드를 종료합니다.

```
exit
```

단계 22 시스템 모드로 들어갑니다.

```
scopesystem
```



단계 23 서비스 모드로 들어갑니다.

**scopeservices**

단계 24 새 인증서를 사용하도록 FXOS 웹 서비스를 구성합니다.

**sethttpskeyringkeyring\_name**

단계 25 컨피그레이션을 커밋합니다.

**commit-buffer**

단계 26 HTTPS 서버와 연결된 키 링을 표시합니다. 이 절차의 3단계에서 생성한 키 링 이름을 반영해야 합니다. 화면 출력에 기본 키 링 이름이 표시되면 HTTPS 서버가 아직 새 인증서를 사용하도록 업데이트 되지 않은 것입니다.

**showhttps**

예제:

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

단계 27 가져온 인증서의 내용을 표시하고, **Certificate Status** 값이 **Valid**로 표시되는지 확인합니다.

**scopesecurity**

**showkeyringkeyring\_namedetail**

예제:

```
fp4120 /security # scope security
fp4120 /security # show keyring firepower_cert detail
Keyring firepower_cert:
  RSA key modulus: Mod2048
  Trustpoint CA: firepower_chain
  Certificate status: Valid
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number:
    45:00:00:00:0a:de:86:55:16:82:24:f3:be:00:00:00:00:00:0a
  Signature Algorithm: ecdsa-with-SHA256
  Issuer: DC=local, DC=naaustin, CN=naaustin-NAAUSTIN-PC-CA
  Validity
    Not Before: Apr 28 13:09:54 2016 GMT
    Not After : Apr 28 13:09:54 2018 GMT
  Subject: C=US, ST=California, L=San Jose, O=Cisco Systems, OU=TAC,
CN=fp4120.test.local
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    Modulus:
      00:b3:43:8d:e6:06:a0:91:f6:76:7e:2e:09:54:40:
      0d:1b:ee:d8:1a:07:07:6e:75:2d:ec:ba:55:c8:c0:
      a1:9c:a3:8f:26:30:70:91:43:0d:40:0d:66:42:c4:
      50:0d:c6:c9:db:7d:bf:b0:ad:1f:31:29:f2:a8:e2:
      fc:27:30:bb:8a:dc:5e:54:46:51:cb:3e:ff:6b:1e:
      d6:18:db:de:83:f5:cf:fb:37:74:de:7b:78:19:73:
      3a:dc:5b:2b:3d:c3:e6:03:b1:30:82:a0:d2:2e:84:
      a1:b4:11:15:d7:48:61:7f:8f:8d:c3:8a:4a:09:9f:
      9e:49:29:12:26:44:c1:d5:91:da:29:5f:5b:b6:d6:
```



다음에 할 작업

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



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

## 소프트웨어 장애에서 복구

시작하기 전에

시스템의 성공적인 부팅을 방해하는 소프트웨어 장애가 발생하면 다음 절차에 따라 소프트웨어의 새 버전을 부팅할 수 있습니다. 이 프로세스를 완료하려면 키스타트 이미지를 TFTP 부팅하고, 새 시스템과 관리자 이미지를 다운로드하고, 새 이미지를 사용하여 부팅해야 합니다.

Cisco.com의 다음 위치에서 특정 FXOS 버전에 대한 복구 이미지를 가져올 수 있습니다.

- Firepower 9300—<https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286287252&flowid=77282&softwareid=286287263>
- Firepower 4100 Series—<https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286305187&flowid=79423&softwareid=286287263>

복구 이미지에는 세 개의 별도 파일이 포함되어 있습니다. 예를 들어 다음은 FXOS 2.1.1.64의 현재 복구 이미지입니다.

```
Recovery image (kickstart) for FX-OS 2.1.1.64.
fxos-k9-kickstart.5.0.3.N2.4.11.63.SPA
```

```
Recovery image (manager) for FX-OS 2.1.1.64.
fxos-k9-manager.4.1.1.63.SPA
```

```
Recovery image (system) for FX-OS 2.1.1.64.
fxos-k9-system.5.0.3.N2.4.11.63.SPA
```

절차

**단계 1** ROMMON에 액세스합니다.

- 콘솔 포트에 연결합니다.
- 시스템을 재부팅합니다.  
시스템이 로딩을 시작하며, 로딩 프로세스 중에 카운트다운 타이머가 표시됩니다.
- 카운트다운 중에 **Escape** 키를 눌러 ROMMON 모드로 들어갑니다.

## 예제:

```

Cisco System ROMMON, version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015 by Cisco Systems, Inc.
Compiled Sun 01/01/1999 23:59:59.99 by user

Current image running: Boot ROM0
Last reset cause: LocalSoft
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address aa:aa:aa:aa:aa:aa

find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.0.00.00.SPA

bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.0.00.00.SPA

Use BREAK, ESC or CTRL+L to interrupt boot.
use SPACE to begin boot immediately.
Boot interrupted.

rommon 1 >

```

## 단계 2 킥스타트 이미지를 TFTP 부팅합니다.

- a) 관리 IP 주소, 관리 넷마스크, 게이트웨이 IP 주소가 올바르게 설정되었는지 확인합니다. **set** 명령을 사용하여 해당 값을 볼 수 있습니다. **ping** 명령을 사용하여 TFTP 서버에 대한 연결을 테스트할 수 있습니다.

```

rommon 1 > set
ADDRESS=
NETMASK=
GATEWAY=
SERVER=
IMAGE=
PS1="ROMMON ! > "
rommon > address <ip-address>
rommon > netmask <network-mask>
rommon > gateway <default-gateway>

```

- b) 킥스타트 이미지를 FXOS 새시에서 액세스 가능한 TFTP 디렉터리에 복사합니다.  
참고 킥스타트 이미지 버전 번호는 번들 버전 번호와 일치하지 않습니다. Cisco.com 소프트웨어 다운로드 페이지에서 FXOS 버전과 킥스타트 이미지 간 매핑을 보여주는 정보를 찾을 수 있습니다.
- c) boot 명령을 사용하여 ROMMON에서 이미지를 부팅합니다.

```

boot tftp://<IP address>/<path to image>

```

참고 FXOS 새시의 전면 패널에 있는 USB 슬롯에 삽입한 USB 미디어 디바이스를 사용하여 ROMMON에서 킥스타트를 부팅할 수도 있습니다. 시스템이 실행 중일 때 USB 디바이스를 삽입하는 경우 시스템을 리부팅해야 USB 디바이스가 인식됩니다. 이미지가 수신 중임을 나타내는 일련의 # 표시가 나타난 다음 킥스타트 이미지가 로드됩니다.

## 예제:

```

rommon 1 > set
ADDRESS=
NETMASK=
GATEWAY=

```

```

SERVER=
IMAGE=
PS1="ROMMON ! > "

rommon 2 > address 10.0.0.2
rommon 3 > netmask 255.255.255.0
rommon 4 > gateway 10.0.0.1
rommon 5 > ping 10.0.0.2
..!!!!!!
Success rate is 100 percent (10/10)
rommon 6 > ping 192.168.1.2
..!!!!!!
Success rate is 100 percent (10/10)

rommon 7 > boot tftp://192.168.1.2/fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA
ADDRESS: 10.0.0.2
NETMASK: 255.255.255.0
GATEWAY: 10.0.0.1
SERVER: 192.168.1.2
IMAGE: fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA

TFTP_MACADDR: aa:aa:aa:aa:aa:aa
.....
Receiving fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA from 192.168.1.2

#####
#####
#####

File reception completed.
    
```

단계 3 FXOS 새시에 방금 로드한 킥스타트 이미지와 일치하는 복구 시스템 및 관리자 이미지를 다운로드합니다.

a) 복구 시스템 및 관리자 이미지를 다운로드하려면 관리 IP 주소 및 게이트웨이를 설정해야 합니다. USB를 통해 이러한 이미지를 다운로드할 수 없습니다.

```

switch(boot) # config terminal
switch(boot) (config) # interface mgmt 0
switch(boot) (config-if) # ip address <ip address> <netmask>
switch(boot) (config-if) # no shutdown
switch(boot) (config-if) # exit
switch(boot) (config) # ip default-gateway <gateway>
switch(boot) (config) # exit
    
```

b) 원격 서버에서 bootflash로 복구 시스템 및 관리자 이미지를 복사합니다.

switch(boot)# copy URL bootflash:

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- ftp://username@hostname/path/image\_name
- scp://username@hostname/path/image\_name
- sftp://username@hostname/path/image\_name
- tftp://hostname/path/image\_name

예제:

```

switch(boot) # copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
bootflash:
    
```

```
switch(boot) # copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
bootflash:
```

- c) 이미지를 성공적으로 FXOS 재시에 복사한 후 `nuova-sim-mgmt-nsg.0.1.0.001.bin`에서 관리자 이미지로 `symlink`를 만듭니다. 이 링크는 로드할 관리자 이미지를 로드 메커니즘에 알려줍니다. 어떤 이미지를 로드하려고 하는지와 상관없이 `symlink` 이름은 항상 `nuova-sim-mgmt-nsg.0.1.0.001.bin`이어야 합니다.

```
switch(boot) # copy bootflash:<manager-image>
bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin
```

예제:

```
switch(boot) # config terminal
Enter configuration commands, one per line. End with CNTL/Z.

switch(boot) (config) # interface mgmt 0
switch(boot) (config-if) # ip address 10.0.0.2 255.255.255.0
switch(boot) (config-if) # no shutdown
switch(boot) (config-if) # exit
switch(boot) (config) # ip default-gateway 10.0.0.1
switch(boot) (config) # exit
switch(boot) # copy
  tftp://192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
bootflash:
Trying to connect to tftp server.....
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot) # copy
  tftp://192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
bootflash:
Trying to connect to tftp server.....
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot) # copy bootflash:fxos-k9-manager.4.1.1.69.SPA
bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin

Copy complete, now saving to disk (please wait)...

switch(boot) #
```

- 단계 4 방금 다운로드한 시스템 이미지를 로드합니다.

```
switch(boot) # load bootflash:<system-image>
```

예제:

```
switch(boot) # load bootflash:fxos-k9-system.5.0.3.N2.4.11.69.SPA
Uncompressing system image: bootflash:/fxos-k9-system.5.0.3.N2.4.11.69.SPA

Manager image digital signature verification successful
...
System is coming up ... Please wait ...

Cisco FPR Series Security Appliance
```

FP9300-A login:

**단계 5** 시스템이 이전 이미지를 로드하려고 시도하지 못하게 하려면, 복구 이미지를 로드한 후 다음 명령을 입력합니다.

참고 이 단계는 복구 이미지를 로드한 직후 수행해야 합니다.

```
FP9300-A# scope org
FP9300-A /org # scope fw-platform-pack default
FP9300-A /org/fw-platform-pack # set platform-bundle-version ""
Warning: Set platform version to empty will result software/firmware incompatibility issue.
FP9300-A /org/fw-platform-pack* # commit-buffer
```

**단계 6** FXOS 새시에서 사용할 플랫폼 번들 이미지를 다운로드 및 설치합니다. 자세한 내용은 [이미지 관리, 47 페이지](#)를 참고하십시오.

예제:

```
FP9300-A# scope firmware
FP9300-A /firmware # show download-task

Download task:
-----
File Name Protocol Server          Port      Userid      State
-----
fxos-k9.2.1.1.73.SPA
      Tftp      192.168.1.2          0          Downloaded
FP9300-A /firmware # show package fxos-k9.2.1.1.73.SPA detail
Firmware Package fxos-k9.2.1.1.73.SPA:
  Version: 2.1(1.73)
  Type: Platform Bundle
  State: Active
Time Stamp: 2012-01-01T07:40:28.000
Build Date: 2017-02-28 13:51:08 UTC
FP9300-A /firmware #
```

## 손상된 파일 시스템에서 복구

시작하기 전에

Supervisor의 온보드 플래시가 손상되고 시스템을 더 이상 성공적으로 시작할 수 없는 경우 다음 절차를 사용하여 시스템을 복구할 수 있습니다. 이 프로세스를 완료하려면 킥스타트 이미지를 TFTP 부팅하고, 플래시를 재포맷하고, 새 시스템과 관리자 이미지를 다운로드하고, 새 이미지를 사용하여 부팅해야 합니다.



참고 이 절차에는 시스템 플래시 재포맷이 포함됩니다. 그 결과, 시스템이 복구된 후 완전히 다시 구성해야 합니다.

Cisco.com의 다음 위치에서 특정 FXOS 버전에 대한 복구 이미지를 가져올 수 있습니다.

- Firepower 9300—<https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286287252&flowid=77282&softwareid=286287263>

- Firepower 4100 Series—<https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286305187&flowid=79423&softwareid=286287263>

복구 이미지에는 세 개의 별도 파일이 포함되어 있습니다. 예를 들어 다음은 FXOS 2.1.1.64의 복구 이미지입니다.

```
Recovery image (kickstart) for FX-OS 2.1.1.64.
fxos-k9-kickstart.5.0.3.N2.4.11.63.SPA
```

```
Recovery image (manager) for FX-OS 2.1.1.64.
fxos-k9-manager.4.1.1.63.SPA
```

```
Recovery image (system) for FX-OS 2.1.1.64.
fxos-k9-system.5.0.3.N2.4.11.63.SPA
```

## 절차

**단계 1** ROMMON에 액세스합니다.

- 콘솔 포트에 연결합니다.
- 시스템을 재부팅합니다.  
시스템이 로딩을 시작하며, 로딩 프로세스 중에 카운트다운 타이머가 표시됩니다.
- 카운트다운 중에 **Escape** 키를 눌러 ROMMON 모드로 들어갑니다.

예제:

```
Cisco System ROMMON, version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015 by Cisco Systems, Inc.
Compiled Sun 01/01/1999 23:59:59.99 by user
```

```
Current image running: Boot ROM0
Last reset cause: LocalSoft
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!
```

```
Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address aa:aa:aa:aa:aa:aa
```

```
find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.0.00.00.SPA
bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.0.00.00.SPA
```

```
Use BREAK, ESC or CTRL+L to interrupt boot.
use SPACE to begin boot immediately.
Boot interrupted.
```

```
rommon 1 >
```

**단계 2** 키스타트 이미지를 TFTP 부팅합니다.

- 관리 IP 주소, 관리 넷마스크, 게이트웨이 IP 주소가 올바르게 설정되었는지 확인합니다. **set** 명령을 사용하여 해당 값을 볼 수 있습니다. **ping** 명령을 사용하여 TFTP 서버에 대한 연결을 테스트할 수 있습니다.

```
rommon 1 > set
ADDRESS=
NETMASK=
GATEWAY=
SERVER=
```



```

IMAGE=
PS1="ROMMON ! > "
rommon > address <ip-address>
rommon > netmask <network-mask>
rommon > gateway <default-gateway>
    
```

- b) 킥스타트 이미지를 FXOS 새시에서 액세스 가능한 TFTP 디렉터리에 복사합니다.  
참고 킥스타트 이미지 버전 번호는 번들 버전 번호와 일치하지 않습니다. Cisco.com 소프트웨어 다운로드 페이지에서 FXOS 버전과 킥스타트 이미지 간 매핑을 보여주는 정보를 찾을 수 있습니다.

- c) boot 명령을 사용하여 ROMMON에서 이미지를 부팅합니다.

```

boot tftp://<IP address>/<path to image>
    
```

참고 FXOS 새시의 전면 패널에 있는 USB 슬롯에 삽입한 USB 미디어 디바이스를 사용하여 ROMMON에서 킥스타트를 부팅할 수도 있습니다. 시스템이 실행 중일 때 USB 디바이스를 삽입하는 경우 시스템을 리부팅해야 USB 디바이스가 인식됩니다. 이미지가 수신 중임을 나타내는 일련의 # 표시가 나타난 다음 킥스타트 이미지가 로드됩니다.

예제:

```

rommon 1 > set
ADDRESS=
NETMASK=
GATEWAY=
SERVER=
IMAGE=
PS1="ROMMON ! > "

rommon 2 > address 10.0.0.2
rommon 3 > netmask 255.255.255.0
rommon 4 > gateway 10.0.0.1
rommon 5 > ping 10.0.0.2
..!!!!!!
Success rate is 100 percent (10/10)
rommon 6 > ping 192.168.1.2
..!!!!!!
Success rate is 100 percent (10/10)

rommon 7 > boot tftp://192.168.1.2/fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA
ADDRESS: 10.0.0.2
NETMASK: 255.255.255.0
GATEWAY: 10.0.0.1
SERVER: 192.168.1.2
IMAGE: fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA

TFTP_MACADDR: aa:aa:aa:aa:aa:aa
.....
Receiving fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA from 192.168.1.2

#####
#####
#####

File reception completed.
    
```

- 단계 3 킥스타트 이미지가 로드된 후 **init system** 명령을 사용하여 플래시를 재포맷합니다. **init system** 명령은 시스템에 다운로드된 모든 소프트웨어 이미지 및 시스템의 모든 컨피그레이션을 포함하여 플래시의 콘텐츠를 지웁니다. 이 명령을 완료하는 데 약 20~30분 정도 소요됩니다.

예제:

```
switch(boot)# init system
```

This command is going to erase your startup-config, licenses as well as the contents of your bootflash:.

```
Do you want to continue? (y/n) [n] y
```

```
Detected 32GB flash...
Initializing the system
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
Initializing startup-config and licenses
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
Formatting bootflash:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
Formatting SAM partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
Formatting Workspace partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
Formatting Sysdebug partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):           done
```

#### 단계 4 복구 이미지를 FXOS 새시에 다운로드합니다.

- a) 복구 이미지를 다운로드하려면 관리 IP 주소 및 게이트웨이를 설정해야 합니다. USB를 통해 이러한 이미지를 다운로드할 수 없습니다.

```
switch(boot)# config terminal
switch(boot) (config)# interface mgmt 0
switch(boot) (config-if)# ip address <ip address> <netmask>
switch(boot) (config-if)# no shutdown
switch(boot) (config-if)# exit
switch(boot) (config)# ip default-gateway <gateway>
switch(boot) (config)# exit
```

- b) 원격 서버에서 bootflash로 복구 이미지 세 개를 모두 복사합니다.

```
switch(boot)# copy URL bootflash:
```

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp://username@hostname/path/image\_name**
- **scp://username@hostname/path/image\_name**
- **sftp://username@hostname/path/image\_name**
- **tftp://hostname/path/image\_name**

예제:

```
switch(boot)# copy
scp://<username>@192.168.1.2/recovery_images/fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
bootflash:
```

```
switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:
```

```
switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
```

- c) 이미지를 성공적으로 FXOS 새시에 복사한 후 `nuova-sim-mgmt-nsg.0.1.0.001.bin`에서 관리자 이미지로 symlink를 만듭니다. 이 링크는 로드할 관리자 이미지를 로드 메커니즘에 알려줍니다. 어떤 이미지를 로드하려고 하는지와 상관없이 symlink 이름은 항상 `nuova-sim-mgmt-nsg.0.1.0.001.bin`이어야 합니다.

```
switch(boot)# copy bootflash:<manager-image>
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin
```

#### 예제:

```
switch(boot)# config terminal
Enter configuration commands, one per line. End with CNTL/Z.

switch(boot) (config)# interface mgmt 0
switch(boot) (config-if)# ip address 10.0.0.2 255.255.255.0
switch(boot) (config-if)# no shutdown
switch(boot) (config-if)# exit
switch(boot) (config)# ip default-gateway 10.0.0.1
switch(boot) (config)# exit
switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server.....
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server.....
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
Trying to connect to tftp server.....
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy bootflash:fxos-k9-manager.4.1.1.69.SPA
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin

Copy complete, now saving to disk (please wait)...

switch(boot)#
```

#### 단계 5 스위치를 로드합니다.

```
switch(boot)# reload
```

예제:

```
switch(boot)# reload
This command will reboot this supervisor module. (y/n) ? y
[ 1866.310313] Restarting system.

!! Rommon image verified successfully !!

Cisco System ROMMON, Version 1.0.11, RELEASE SOFTWARE
Copyright (c) 1994-2016 by Cisco Systems, Inc.
Compiled Wed 11/23/2016 11:23:23.47 by builder
Current image running: Boot ROM1
Last reset cause: ResetRequest
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!
BIOS has been locked !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address: bb:aa:77:aa:aa:bb

autoboot: Can not find autoboot file 'menu.lst.local'
Or can not find correct boot string !!
rommon 1 >
```

단계 6 킥스타트 및 시스템 이미지에서 부팅합니다.

```
rommon 1 > boot <kickstart-image> <system-image>
```

참고 시스템 이미지가 로드되는 동안 라이선스 관리자 실패 메시지가 표시됩니다. 이러한 메시지는 안전하게 무시할 수 있습니다.

예제:

```
rommon 1 > dir
Directory of: bootflash:\

01/01/12 12:33a <DIR>          4,096 .
01/01/12 12:33a <DIR>          4,096 ..
01/01/12 12:16a <DIR>        16,384 lost+found
01/01/12 12:27a              34,333,696 fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
01/01/12 12:29a              330,646,465 fxos-k9-manager.4.1.1.69.SPA
01/01/12 12:31a              250,643,172 fxos-k9-system.5.0.3.N2.4.11.69.SPA
01/01/12 12:34a              330,646,465 nuova-sim-mgmt-nsg.0.1.0.001.bin
      4 File(s) 946,269,798 bytes
      3 Dir(s)

rommon 2 > boot fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA fxos-k9-system.5.0.3.N2.4.11.69.SPA
!! Kickstart Image verified successfully !!

Linux version: 2.6.27.47 (security@cisco.com) #1 SMP Thu Nov 17 18:22:00 PST 2016
[ 0.000000] Fastboot Memory at 0c100000 of size 201326592
Usage: init 0123456SsQqAaBbCcUu

INIT: version 2.86 booting

POST INIT Starts at Sun Jan  1 00:27:32 UTC 2012
S10mount-ramfs.supnuovaca Mounting /isan 3000m
Mounted /isan
Creating /callhome..
Mounting /callhome..
Creating /callhome done.
Callhome spool file system init done.
Platform is BS or QP MIO: 30
FPGA Version 0x00010500 FPGA Min Version 0x00000600
Checking all filesystems..r.r..r done.
Warning: switch is starting up with default configuration
Checking NVRAM block device ... done
```

```
.
FIPS power-on self-test passed
Unpack CMC Application software
Loading system software
Uncompressing system image: bootflash:/fxos-k9-system.5.0.3.N2.4.11.69.SPA
```

```
Manager image digital signature verification successful
```

```
...
```

```
System is coming up ... Please wait ...
nohup: appending output to `nohup.out'
```

```
---- Basic System Configuration Dialog ----
```

```
This setup utility will guide you through the basic configuration of
the system. Only minimal configuration including IP connectivity to
the Fabric interconnect and its clustering mode is performed through these steps.
```

```
Type Ctrl-C at any time to abort configuration and reboot system.
To back track or make modifications to already entered values,
complete input till end of section and answer no when prompted
to apply configuration.
```

```
You have chosen to setup a new Security Appliance. Continue? (y/n):
```

단계 7 이미지가 로드되면 초기 컨피그레이션 설정을 입력하라는 프롬프트가 표시됩니다. 자세한 내용은 [초기 구성, 10 페이지](#)를 참고하십시오.

단계 8 FXOS 새시에서 사용할 플랫폼 번들 이미지를 다운로드합니다. 플랫폼 번들 이미지 버전은 시스템 복구에 사용한 이미지와 일치해야 합니다. 자세한 내용은 [이미지 관리, 47 페이지](#)를 참고하십시오.

예제:

```
FP9300-A# scope firmware
FP9300-A /firmware # show download-task
```

```
Download task:
```

File Name	Protocol	Server	Port	Userid	State
fxos-k9.2.1.1.73.SPA	Tftp	192.168.1.2	0		Downloaded

```
FP9300-A /firmware # show package fxos-k9.2.1.1.73.SPA detail
Firmware Package fxos-k9.2.1.1.73.SPA:
  Version: 2.1(1.73)
  Type: Platform Bundle
  State: Active
Time Stamp: 2012-01-01T07:40:28.000
Build Date: 2017-02-28 13:51:08 UTC
FP9300-A /firmware #
```

단계 9 플랫폼 번들을 성공적으로 다운로드한 후에는, 나중에 시스템을 로드할 때 사용할 수 있도록 킥스타트 및 시스템 이미지를 수동으로 활성화해야 합니다. 실행 중인 버전이 제안된 시작 버전과 일치하므로, 이 절차를 사용하여 손상된 파일 시스템에서 복구할 때는 자동 활성화가 발생하지 않습니다.

a) 패브릭 인터커넥트 a의 범위를 설정합니다.

```
FP9300-A# scope fabric-interconnect a
```

b) 실행 중인 커널 버전 및 실행 중인 시스템 버전을 보려면 **show version** 명령을 사용합니다. 이러한 문자열을 사용하여 이미지를 활성화합니다.

```
FP9300-A /fabric-interconnect # show version
```

- c) 다음 명령을 입력하여 이미지를 활성화합니다.

```
FP9300-A /fabric-interconnect # activate firmware
      kernel-version <running_kernel_version> system-version <running_system_version>
commit-buffer
```

참고 서버 상태가 "Disk Failed(디스크 실패)"로 변경될 수 있습니다. 이 메시지에 대해 걱정할 필요가 없으며 이 절차를 계속 진행할 수 있습니다.

- d) 시작 버전이 올바르게 설정되었는지 확인하고 이미지의 활성화 상태를 모니터링하려면 **show version** 명령을 사용합니다.

중요 상태가 "Activating(활성)"에서 "Ready(준비)"로 변경될 때까지 다음 단계로 진행하지 마십시오.

```
FP9300-A /fabric-interconnect # show version
```

예제:

```
FP9300-A /firmware # top
FP9300-A# scope fabric-interconnect a
FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
  Running-Kern-Vers: 5.0(3)N2(4.11.69)
  Running-Sys-Vers: 5.0(3)N2(4.11.69)
  Package-Vers: 2.1(1.73)
  Startup-Kern-Vers:
  Startup-Sys-Vers:
  Act-Kern-Status: Ready
  Act-Sys-Status: Ready
  Bootloader-Vers:

FP9300-A /fabric-interconnect # activate firmware kernel-version
5.0(3)N2(4.11.69) system-version 5.0(3)N2(4.11.69)
Warning: When committed this command will reset the end-point
FP9300-A /fabric-interconnect* # commit-buffer
FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
  Running-Kern-Vers: 5.0(3)N2(4.11.69)
  Running-Sys-Vers: 5.0(3)N2(4.11.69)
  Package-Vers: 2.1(1.73)
  Startup-Kern-Vers: 5.0(3)N2(4.11.69)
  Startup-Sys-Vers: 5.0(3)N2(4.11.69)
  Act-Kern-Status: Activating
  Act-Sys-Status: Activating
  Bootloader-Vers:

FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
  Running-Kern-Vers: 5.0(3)N2(4.11.69)
  Running-Sys-Vers: 5.0(3)N2(4.11.69)
  Package-Vers: 2.1(1.73)
  Startup-Kern-Vers: 5.0(3)N2(4.11.69)
  Startup-Sys-Vers: 5.0(3)N2(4.11.69)
  Act-Kern-Status: Ready
  Act-Sys-Status: Ready
  Bootloader-Vers:
```

단계 10 시스템을 재부팅합니다.

예제:

```
FP9300-A /fabric-interconnect # top
FP9300-A# scope chassis 1
FP9300-A /chassis # reboot no-prompt
Starting chassis reboot. Monitor progress with the command "show fsm status"
```

FP9300-A /chassis #

시스템은 각 보안 모듈/엔진의 전원을 끈 다음 마지막으로 FXOS 새시의 전원을 끄고 다시 시작합니다. 이 프로세스는 약 5~10분 정도 걸립니다.

**단계 11** 시스템 상태를 모니터링합니다. 서버 상태가 "Discovery(검색)"에서 "Config(구성)"로 바뀐 다음 마지막으로 "Ok"로 바뀝니다.

예제:

FP9300-A# show server status

Server	Slot	Status	Overall Status	Discovery
1/1		Equipped	Discovery	In Progress
1/2		Equipped	Discovery	In Progress
1/3		Empty		

FP9300-A# show server status

Server	Slot	Status	Overall Status	Discovery
1/1		Equipped	Config	Complete
1/2		Equipped	Config	Complete
1/3		Empty		

FP9300-A# show server status

Server	Slot	Status	Overall Status	Discovery
1/1		Equipped	Ok	Complete
1/2		Equipped	Ok	Complete
1/3		Empty		

Overall Status(전체 상태)가 "Ok"이면 시스템이 복구된 것입니다. 여전히 보안 어플라이언스를 재구성하고(라이선스 컨피그레이션 포함) 논리적 디바이스를 다시 생성해야 합니다. 자세한 내용:

- Firepower 9300 빠른 시작 가이드—<http://www.cisco.com/go/firepower9300-quick>
- Firepower 9300 환경 설정 가이드—<http://www.cisco.com/go/firepower9300-config>
- Firepower 4100 Series 빠른 시작 가이드—<http://www.cisco.com/go/firepower4100-quick>
- Firepower 4100 Series 환경 설정 가이드—<http://www.cisco.com/go/firepower4100-config>







## 플랫폼 설정

- 날짜 및 시간 설정, 103 페이지
- SSH 구성, 108 페이지
- 텔넷 구성, 109 페이지
- SNMP 구성, 110 페이지
- HTTPS 구성, 117 페이지
- AAA 구성, 128 페이지
- Syslog 구성, 138 페이지
- DNS 서버 구성, 140 페이지

## 날짜 및 시간 설정

시스템에서 NTP(network time protocol)를 구성하거나, 수동으로 날짜 및 시간을 설정하거나, 현재 시스템 시간을 보려면 아래에 설명된 CLI 명령을 사용하십시오.

NTP 설정은 FXOS 새시 및 새시에 설치된 논리적 디바이스 간에 자동으로 동기화됩니다.



참고

FXOS 새시에 Firepower Threat Defense를 구축할 경우, Smart Licensing의 올바른 작동 및 디바이스 등록 시 올바른 타임스탬프를 보장하려면 FXOS 새시에서 NTP를 구성해야 합니다. FXOS 새시 및 Firepower Management Center는에 대해 동일한 NTP 서버를 사용해야 합니다.

NTP를 사용하는 경우 **Current Time**(현재 시간) 탭에서 전반적인 동기화 상태를 볼 수 있습니다. 또는 **Time Synchronization**(시간 동기화) 탭의 **NTP Server**(NTP 서버) 테이블에 있는 **Server Status**(서버 상태) 필드에서 구성된 각 NTP 서버의 동기화 상태를 볼 수 있습니다. 시스템을 특정 NTP 서버와 동기화할 수 없는 경우 **Server Status**(서버 상태) 옆에 있는 정보 아이콘에 마우스 커서를 대면 자세한 내용을 확인할 수 있습니다.

## 구성된 날짜 및 시간 보기

### 절차

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

단계 2 다음 명령을 사용하여 구성된 표준 시간대를 확인합니다.

```
Firepower-chassis# showtimezone
```

단계 3 구성된 날짜 및 시간을 보려면:

```
Firepower-chassis# showclock
```

다음 예제는 구성된 시간대 및 현재 시스템 날짜 및 시간을 표시하는 방법을 보여줍니다.

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

## 표준 시간대 설정

### 절차

단계 1 시스템 모드를 입력합니다.

```
Firepower-chassis# scope system
```

단계 2 시스템 서비스 모드를 입력합니다.

```
Firepower-chassis /system # scope services
```

단계 3 표준 시간대를 설정합니다.

```
Firepower-chassis /system/services # set timezone
```

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

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

단계 4 다음 명령을 사용하여 구성된 표준 시간대를 확인합니다.

```
Firepower-chassis /system/services # top
```

```
Firepower-chassis# show timezone
```

다음의 예에서는 표준 시간대를 태평양 표준 시간대로 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#

```

## NTP를 사용하여 날짜 및 시간 설정

NTP는 네트워크 시스템 간에 정확하게 동기화된 시간을 제공하는 계층적 서버 시스템을 구현하는 데 사용됩니다. 정밀한 타임 스탬프가 포함된 CRL 검증과 같이 시간에 민감한 작업에는 이러한 정확성이 필요합니다. 최대 4개까지 NTP 서버를 구성할 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 지정된 호스트 이름, IPv4 또는 IPv6 주소가 있는 NTP 서버를 사용하도록 시스템을 구성합니다.  
Firepower-chassis /system/services # **create ntp-server {hostname | ip-addr | ip6-addr}**
  - 단계 4 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
Firepower-chassis /system/services # **commit-buffer**
  - 단계 5 구성된 모든 NTP 서버의 동기화 상태를 보려면:  
Firepower-chassis /system/services # **show ntp-server**
  - 단계 6 특정 NTP 서버의 동기화 상태를 보려면:  
Firepower-chassis /system/services # **scopentp-server{hostname | ip-addr | ip6-addr}**  
Firepower-chassis /system/services/ntp-server # **show detail**
- 

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

```

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

```

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

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

## NTP 서버 삭제

### 절차

- 
- 단계 1** 시스템 모드를 입력합니다.  
Firepower-chassis# **scope system**
- 단계 2** 시스템 서비스 모드를 입력합니다.  
Firepower-chassis /system # **scope services**
- 단계 3** 지정된 호스트 이름, IPv4 또는 IPv6 주소가 있는 NTP 서버를 삭제합니다.  
Firepower-chassis /system/services # **delete ntp-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 새시에 날짜 및 시간을 수동으로 설정하는 방법을 설명합니다. 시스템 클록 수정사항은 즉시 적용됩니다.



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

---

## 절차

단계 1 시스템 모드를 입력합니다.

```
Firepower-chassis# scope system
```

단계 2 시스템 서비스 모드를 입력합니다.

```
Firepower-chassis /system # scope services
```

단계 3 시스템 클록을 구성합니다.

```
Firepower-chassis /system/services # set clock month day year hour min sec
```

월의 경우, 월의 첫 세 자릿수를 사용합니다. 시간은 24시간 형식으로 입력해야 하며 이때 7pm은 19로 입력합니다.

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 새시에 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 새시에 대한 SSH 액세스를 구성하려면 다음 중 하나를 수행합니다.

- Firepower 새시에 대한 SSH 액세스를 허용하려면 다음 명령을 입력합니다.  

```
Firepower-chassis /system/services # enable ssh-server
```
- Firepower 새시에 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 새시에 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 새시에 대한 텔넷 액세스를 활성화하거나 비활성화하는 방법을 설명합니다. 텔넷은 기본적으로 비활성화되어 있습니다.



참고

텔넷 컨피그레이션은 현재 CLI를 사용하는 경우에만 사용할 수 있습니다.

### 절차

- 단계 1 시스템 모드를 입력합니다.  
Firepower-chassis #**scope system**
- 단계 2 시스템 서비스 모드를 입력합니다.  
Firepower-chassis /system #**scope services**
- 단계 3 Firepower 새시에 대한 텔넷 액세스를 구성하려면 다음 중 하나를 수행합니다.
  - Firepower 새시에 대한 텔넷 액세스를 허용하려면 다음 명령을 입력합니다.  
Firepower-chassis /system/services # **enable telnet-server**
  - Firepower 새시에 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 새시에 SNMP(Simple Network Management Protocol)를 구성하는 방법을 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

## SNMP 정보

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

SNMP 프레임워크는 세 부분으로 구성됩니다.

- SNMP 관리자—SNMP를 사용하는 네트워크 디바이스의 활동을 제어하고 모니터링하는 데 쓰이는 시스템.
- SNMP 에이전트 — Firepower 새시 데이터를 유지 관리하고 필요 시 데이터를 SNMP 관리자에 보고하는 Firepower 새시에 포함된 소프트웨어 구성 요소입니다. Firepower 새시는 MIB 컬렉션 및 에이전트를 포함합니다. SNMP 에이전트를 활성화하고 관리자 및 에이전트 간 관계를 생성하려면 Firepower Chassis Manager 또는 FXOS CLI에서 SNMP를 활성화하고 구성합니다.
- MIB(managed information base) - SNMP 에이전트에 있는 관리되는 개체의 모음.

Firepower 새시는 SNMPv1, SNMPv2c 및 SNMPv3를 지원합니다. SNMPv1 및 SNMPv2c 둘 다 커뮤니티 기반 보안 유형을 사용합니다. SNMP는 다음과 같이 정의됩니다.

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



## SNMP 알림

SNMP의 핵심 기능 중 하나는 SNMP 에이전트가 보내는 알림을 생성하는 것입니다. 이러한 알림은 SNMP 관리자가 요청을 보낼 필요 없습니다. 알림은 잘못된 사용자 인증, 재시작, 연결 종료, 네이브 라우터와의 연결 끊김, 기타 중대한 이벤트를 나타낼 수 있습니다.

Firepower 새시는 트랩 또는 알림 중 하나로 SNMP 알림을 생성합니다. 트랩은 SNMP 관리자가 트랩을 수신할 때 승인을 전송하지 않기 때문에 알림보다 신뢰성이 떨어지며 Firepower 새시는 트랩 수신 여부를 확인할 수 없습니다. inform 요청을 수신한 SNMP 관리자는 SNMP 응답 PDU(protocol data unit)로 메시지를 승인합니다. Firepower 새시가 PDU를 수신하지 못하는 경우 알림 요청을 다시 전송할 수 있습니다.

## SNMP 보안 레벨 및 권한

SNMPv1, SNMPv2c, SNMPv3 각각은 서로 다른 보안 모델을 나타냅니다. 보안 모델 및 선택된 보안 레벨의 조합을 통해 SNMP 메시지 처리 시 적용할 보안 메커니즘을 결정합니다.

보안 레벨은 SNMP 트랩과 연결된 메시지를 보는 데 필요한 권한을 결정합니다. 권한 레벨은 메시지가 공개되지 않도록 보호하거나 인증해야 할지 결정합니다. 지원되는 보안 레벨은 어떤 보안 모델이 구현되었는지에 따라 달라집니다. SNMP 보안 레벨은 다음 중 하나 이상의 권한 이상을 지원합니다.

- noAuthNoPriv—인증 또는 암호화 없음
- authNoPriv—인증은 있지만 암호화 없음
- authPriv—인증 및 암호화

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

## SNMP 보안 모델 및 레벨의 지원되는 조합

다음 표에서는 보안 모델 및 레벨의 조합에 대해 설명합니다.

표 8: SNMP 보안 모델 및 레벨

모델	레벨	인증	암호화	결과
v1	noAuthNoPriv	커뮤니티 문자열	없음	인증에 커뮤니티 문자열 매치를 사용합니다.
v2c	noAuthNoPriv	커뮤니티 문자열	없음	인증에 커뮤니티 문자열 매치를 사용합니다.

모델	레벨	인증	암호화	결과
v3	noAuthNoPriv	Username	없음	인증에 사용자 이름 매치를 사용합니다.
v3	authNoPriv	HMAC-SHA	없음	HMAC SHA(Secure Hash Algorithm) 기반 인증을 제공합니다.
v3	authPriv	HMAC-SHA	DES	HMAC-SHA 알고리즘 기반 인증을 제공합니다. CBC(Cipher Block Chaining) DES(DES-56) 표준 기반의 인증과 함께 DES(Data Encryption Standard) 56비트 암호화도 제공합니다.

## SNMPv3 보안 기능

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

- 메시지 통합 - 메시지가 무단으로 변경 또는 손상되지 않았는지, 그리고 데이터 시퀀스가 비약의적인 방식으로 발생할 수 있는 것보다 더 많이 변경되지 않았는지 확인합니다.
- 메시지 출처 인증 - 수신 데이터를 만든 사용자의 클레임된 ID가 확인되도록 보장합니다.
- 메시지 기밀성 및 암호화 - 권한이 없는 개인, 엔티티 또는 프로세스에 정보가 노출 또는 사용되지 않도록 합니다.

## SNMP 지원

Firepower 새시는 SNMP에 다음을 지원합니다.

**MIB** 지원

Firepower 새시는 MIB에 대해 읽기 전용 액세스를 지원합니다.

**SNMPv3** 사용자의 인증 프로토콜

Firepower 새시는 SNMPv3 사용자에게 대해 HMAC-SHA-96(SHA) 인증 프로토콜을 지원합니다.

**SNMPv3** 사용자를 위한 **AES** 프라이버시 프로토콜

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

프라이버시 비밀번호, 즉 `priv` 옵션에서는 SNMP 보안 암호화를 위해 DES 또는 128비트 AES 암호화를 선택할 수 있습니다. AES-128 컨피그레이션을 활성화하고 SNMPv3 사용자에게 대한 프라이버시 비밀번호가 있는 경우, Firepower 새시는 해당 프라이버시 비밀번호를 사용하여 128비트 AES 키를 생성합니다. AES 프라이버시 비밀번호는 8자 이상입니다. 비밀번호가 일반 텍스트로 지정될 경우 최대 64자로 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 입력한 후 SNMP 커뮤니티를 입력하라는 프롬프트가 표시됩니다.
- 단계 4** SNMP 커뮤니티를 지정합니다. 커뮤니티 이름을 비밀번호로 사용합니다. 커뮤니티 이름은 최대 32자의 영숫자 문자열이 될 수 있습니다.  
Firepower-chassis /monitoring # **Enter a snmp community: community-name**
- 단계 5** SNMP를 책임지는 시스템 담당자를 지정합니다. 시스템 연락처 이름은 이메일 주소 또는 이름과 전화번호로, 최대 255자의 영숫자 문자열이 될 수 있습니다.  
Firepower-chassis /monitoring # **set snmp syscontact system-contact-name**
- 단계 6** SNMP 에이전트(서버)가 실행되는 호스트의 위치를 지정합니다. 시스템 위치 이름은 최대 512자의 영숫자 문자열이 될 수 있습니다.  
Firepower-chassis /monitoring # **set snmp syslocation system-location-name**
- 단계 7** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
Firepower-chassis /monitoring # **commit-buffer**
-

다음의 예에서는 SNMP를 활성화하고 SNMP 커뮤니티 SnmpCommSystem2를 구성하고 시스템 담당자 contactperson을 구성하고 연락처 위치 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}
- 가능한 값은 다음과 같습니다.
- 버전으로 v2c 또는 v3를 선택할 경우 **traps**.
  - 버전으로 v2c를 선택할 경우 **informs**.
- 참고 알림은 버전으로 v2c를 선택한 경우에만 전송될 수 있습니다.
- 단계 8 (선택 사항) 버전을 v3로 선택한 경우 트랩과 연결된 권한을 지정합니다.  
Firepower-chassis /monitoring/snmp-trap # **set v3privilege** {auth | noauth | priv}

가능한 값은 다음과 같습니다.

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

**단계 9** 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
Firepower-chassis /monitoring/snmp-trap # commit-buffer
```

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

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

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

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

## SNMP 트랩 삭제

절차

**단계 1** 모니터링 모드를 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 192.168.100.112의 SNMP 트랩을 삭제하고 트랜잭션을 커밋합니다.

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

## SNMPv3 사용자 생성

절차

---

단계 1 모니터링 모드를 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 프롬프트가 표시됩니다.

단계 4 AES-128 암호화 사용을 활성화 또는 비활성화합니다.

```
Firepower-chassis /monitoring/snmp-user # set aes-128 {no | yes}
```

기본적으로 AES-128 암호화는 비활성화되어 있습니다.

단계 5 사용자 프라이버시 비밀번호를 지정합니다.

```
Firepower-chassis /monitoring/snmp-user # set priv-password
```

**set priv-password** 명령을 입력한 후 프라이버시 비밀번호를 입력하고 확인하라는 프롬프트가 표시 됩니다.

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 구성

이 섹션에서는 FXOS 새시에서 SNMP를 구성하는 방법을 설명합니다.



- 
- 참고** Firepower Chassis Manager 또는 FXOS CLI를 사용하여 HTTPS 포트를 변경할 수 있습니다. 다른 모든 HTTPS 컨피그레이션 작업에는 FXOS CLI만 사용해야 합니다.
- 

## 인증서, 키 링, 신뢰 지점

HTTPS에서는 PKI(Public Key Infrastructure)의 구성 요소를 사용하여 두 디바이스, 이클테면 클라이언트 브라우저와 FXOS 새시 간의 보안 통신을 설정합니다.

### 암호화 키 및 키 링

각 PKI 디바이스는 비대칭 RSA(Rivest-Shamir-Adleman) 암호화 키의 쌍을 보유하고 있습니다. 개인 키와 공개 키로 구성된 이 쌍은 내부 키 링에 저장됩니다. 두 키 중 하나로 암호화된 메시지는 나머지 키로 해독할 수 있습니다. 암호화된 메시지를 보낼 때 발신자는 수신자의 공개 키로 메시지를 암호화하며 수신자는 자신의 개인 키로 그 메시지를 해독합니다. 발신자는 알려진 메시지를 자신의 개인 키로 암호화하는 방법으로('서명'이라고도 함) 개인 키에 대한 소유권을 입증할 수도 있습니다. 수신자가 해당 공개 키를 사용하여 성공적으로 메시지를 해독할 수 있다면 발신자가 개인 키를 소유하고 있음이 입증됩니다. 암호화 키의 길이는 다양하지만, 일반적으로 512바이트 ~ 2048바이트입니다. 일반적으로

키가 길수록 더 안전합니다. FXOS에서는 초기 2048비트 키 쌍으로 기본 키 링을 제공하며 사용자가 추가 키 링을 생성할 수 있습니다.

클러스터 이름이 바뀌거나 인증서가 만료될 경우 기본 키 링 인증서를 수동으로 재생성해야 합니다.

### 인증서

안전한 통신을 위해 일차적으로 두 디바이스가 디지털 인증서를 교환합니다. 인증서는 디바이스 공개 키 및 디바이스 ID에 대한 서명된 정보를 포함하는 파일입니다. 디바이스에서 단순히 암호화된 통신을 지원하기 위해서는 자신의 키 쌍 및 자체 서명된 인증서를 생성할 수 있습니다. 원격 사용자가 자체 서명 인증서가 있는 디바이스에 연결할 경우 이 사용자가 디바이스의 ID를 용이하게 확인할 방법이 없으므로 사용자의 브라우저는 초기에 인증 경고를 표시합니다. 기본적으로 FXOS는 기본 키 링의 공개 키를 포함하는 자체 서명 인증서를 기본적으로 갖고 있습니다.

### 신뢰 지점

FXOS에 대한 더 강력한 인증을 제공하기 위해 신뢰할 수 있는 출처 또는 신뢰 지점으로부터 디바이스의 ID를 확인하는 서드파티 인증서를 얻어 설치할 수 있습니다. 서드파티 인증서는 해당 신뢰 지점에서 서명하는데, 이는 루트 CA(certification authority), 중간 CA 또는 루트 CA로 연결되는 신뢰 체인의 일부인 Trust anchor가 될 수 있습니다. 새 인증서를 얻으려면 FXOS를 통해 인증서를 생성하고 신뢰 지점에 요청을 전송해야 합니다.



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

## 키 링 생성

FXOS는 기본 키 링을 포함하여 최대 8개의 키 링을 지원합니다.

### 절차

- 단계 1 보안 모드를 입력합니다.  
Firepower-chassis #**scope security**
- 단계 2 키 링의 이름을 생성합니다.  
Firepower-chassis # **createkeyring***keyring-name*
- 단계 3 SSL 키 길이(비트)를 설정합니다.  
Firepower-chassis # **setmodulus** {**mod1024** | **mod1536** | **mod2048** | **mod512**}
- 단계 4 트랜잭션을 커밋합니다.  
Firepower-chassis # **commit-buffer**

다음 예에서는 키 크기 1024비트의 키 링을 생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create keyring kr220
```



```
Firepower-chassis /security/keyring* # set modulus mod1024
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 할 작업

이 키 링에 대한 인증서 요청을 생성합니다.

## 기본 키 링 재생성

클러스터 이름이 바뀌거나 인증서가 만료될 경우 기본 키 링 인증서를 수동으로 재생성해야 합니다.

절차

- 
- 단계 1 보안 모드를 입력합니다.  
Firepower-chassis #**scope security**
- 단계 2 기본 키 링에 대한 키 링 보안 모드로 들어갑니다.  
Firepower-chassis /security # **scopekeyring default**
- 단계 3 기본 키 링 재생성:  
Firepower-chassis /security/keyring # **setregenerate yes**
- 단계 4 트랜잭션을 커밋합니다.  
Firepower-chassis # **commit-buffer**
- 

다음 예에서는 기본 키 링을 재생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring default
Firepower-chassis /security/keyring* # set regenerate yes
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

## 키 링에 대한 인증서 요청 생성

### 기본 옵션으로 키 링에 대한 인증서 요청 생성

절차

- 
- 단계 1 보안 모드를 입력합니다.  
Firepower-chassis #**scope security**
- 단계 2 키 링에 대한 컨피그레이션 모드로 들어갑니다.  
Firepower-chassis /security # **scope keyringkeyring-name**

단계 3 지정된 IPv4 또는 IPv6 주소 또는 fabric interconnect의 이름을 사용하여 인증서 요청을 만듭니다. 인증서 요청에 대한 비밀번호를 입력하라는 프롬프트가 표시됩니다.

```
Firepower-chassis /security/keyring # create certreq {ip [ipv4-addr | ipv6-v6] |subject-namename}
```

단계 4 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/keyring/certreq # commit-buffer
```

단계 5 Trust anchor 또는 인증 증명으로 복사하여 전송할 수 있는 인증서 요청을 표시합니다.

```
Firepower-chassis /security/keyring # show certreq
```

다음 예는 기본 옵션으로 키 링에 대한 IPv4 주소와 함께 인증서 요청을 만들고 표시합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq ip 192.168.200.123 subject-name sjc04
Certificate request password:
Confirm certificate request password:
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name:
Certificate request country name:
State, province or county (full name):
Locality (eg, city):
Organization name (eg, company):
Organization Unit name (eg, section):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYWljMDQwZGZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF80PhKbhghLAlYZ1F
JqcYEG5Y11+vgohLBTd45s0GC8m4RTLJWHO4SwccAUXQ5Zngf45YtXlWsy1wUWV4
0re/zqTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNiECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHU003Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWICtTWgHhH8BimOb/00KuG8kwfIGGsEDlAv
TTYvUP+Bz9OFiPbRIA718S+V8ndXr1HejiQGx1LDNqoN+odCXPC5kjoXD01zTL09H
BA==
-----END CERTIFICATE REQUEST-----
Firepower-chassis /security/keyring #
```

다음에 할 작업

- BEGIN 및 END 줄을 포함한 인증서 요청의 텍스트를 복사하여 파일에 저장합니다. 키 링에 대한 인증서를 얻을 수 있도록, 인증서 요청이 포함된 파일을 Trust anchor 또는 인증 증명으로 전송합니다.
- 신뢰 지점을 생성하고 Trust anchor로부터 받은 신뢰 인증서에 대한 인증서 체인을 설정합니다.

## 고급 옵션으로 키 링에 대한 인증서 요청 생성

절차

단계 1 보안 모드를 입력합니다.

```
Firepower-chassis #scope security
```

- 단계 2 키 링에 대한 컨피그레이션 모드로 들어갑니다.  
Firepower-chassis /security # **scope keyring***keyring-name*
- 단계 3 인증서 요청을 생성합니다.  
Firepower-chassis /security/keyring # **createcertreq**
- 단계 4 회사가 소재한 국가의 국가 코드를 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set country** *country name*
- 단계 5 요청과 연결된 DNS(Domain Name Server) 주소를 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set dns** *DNS Name*
- 단계 6 인증서 요청과 연결된 이메일 주소를 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set e-mail** *E-mail name*
- 단계 7 FXOS 새시의 IP 주소를 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set ip** {*certificate request ip-address*|*certificate request ip6-address* }
- 단계 8 인증서를 요청하는 회사의 본사가 위치한 시/읍/면을 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set locality** *locality name (eg, city)*
- 단계 9 인증서를 요청하는 조직을 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set org-name** *organization name*
- 단계 10 조직 단위를 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set org-unit-name** *organizational unit name*
- 단계 11 인증서 요청에 대한 비밀번호를 지정합니다(선택 사항).  
Firepower-chassis /security/keyring/certreq\* # **set password** *certificate request password*
- 단계 12 인증서를 요청하는 회사의 본사가 위치한 시/도를 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set state** *state, province or county*
- 단계 13 FXOS 새시의 FQDN(Fully Qualified Domain Name)을 지정합니다.  
Firepower-chassis /security/keyring/certreq\* # **set subject-name** *certificate request name*
- 단계 14 트랜잭션을 커밋합니다.  
Firepower-chassis /security/keyring/certreq # **commit-buffer**
- 단계 15 Trust anchor 또는 인증 증명으로 복사하여 전송할 수 있는 인증서 요청을 표시합니다.  
Firepower-chassis /security/keyring # **show certreq**

다음 예는 고급 옵션으로 키 링에 대한 IPv4 주소와 함께 인증서 요청을 만들고 표시합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq
Firepower-chassis /security/keyring/certreq* # set ip 192.168.200.123
Firepower-chassis /security/keyring/certreq* # set subject-name sjc04
Firepower-chassis /security/keyring/certreq* # set country US
Firepower-chassis /security/keyring/certreq* # set dns bgl-samc-15A
Firepower-chassis /security/keyring/certreq* # set email test@cisco.com
Firepower-chassis /security/keyring/certreq* # set locality new york city
Firepower-chassis /security/keyring/certreq* # set org-name "Cisco Systems"
```

```

Firepower-chassis /security/keyring/certreq* # set org-unit-name Testing
Firepower-chassis /security/keyring/certreq* # set state new york
Firepower-chassis /security/keyring/certreq* # commit-buffer
Firepower-chassis /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name: test@cisco.com
Certificate request country name: US
State, province or county (full name): New York
Locality name (eg, city): new york city
Organization name (eg, company): Cisco
Organization Unit name (eg, section): Testing
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKnlT8qMZO4UGqILKFXQQc2c8b/vW2rnRF80PhKbhghLAlYZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHO4SwccAUXQ5Zngf45YtXlWsyUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBqkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNiECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCcsxN0qUHYGFoQw56RwQueLTNPrndqUwuZHU003Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWlctWgHhH8BimOb/00KuG8kwfIGGsED1Av
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGx1DNqoN+odCXPC5kjoXD01ZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring/certreq #

```

다음에 할 작업

- BEGIN 및 END 줄을 포함한 인증서 요청의 텍스트를 복사하여 파일에 저장합니다. 키 링에 대한 인증서를 얻을 수 있도록, 인증서 요청이 포함된 파일을 Trust anchor 또는 인증 증명으로 전송합니다.
- 신뢰 지점을 생성하고 Trust anchor로부터 받은 신뢰 인증서에 대한 인증서 체인을 설정합니다.

## 신뢰 지점 생성

절차

단계 1 보안 모드를 입력합니다.

```
Firepower-chassis #scope security
```

단계 2 신뢰 지점을 생성합니다.

```
Firepower-chassis /security # createtrustpointname
```

단계 3 이 신뢰 지점에 대한 인증서 정보를 지정합니다.

```
Firepower-chassis /security/trustpoint # setcertchain[ certchain ]
```

명령에 인증서 정보를 지정하지 않은 경우, 루트 CA(Certificate Authority)에 인증 경로를 정의하는 Trust Point 목록 또는 인증서를 입력하라는 프롬프트가 표시됩니다. 해당 정보를 입력한 후 다음 행에 ENDOFBUF를 입력하여 완료합니다.

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

단계 4 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/trustpoint # commit-buffer
```

다음 예에서는 신뢰 지점을 만들고 신뢰 지점에 대한 인증서를 제공합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create trustpoint tPoint10
Firepower-chassis /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGx1IEluYy4xEzARBGNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCCyU
> ZgAMiVyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgd4VBKNOND1
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bf5wZVNAgMBAAGGJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzC190306Mg51zq1zXcz75+VFj2I6rH9asckClD3mkOVx5gJU
> Ptt5CVQpNgNldvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtv1WvfhevskV0j6
> jtceMYZ+f7+3yh421ido3n04MIGeBgNVHSMGgZywgZOAFL1NjtcEMYZ+f7+3yh42
> 1ido3n04oXikdjb0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAct
> C1NhbRrhIENsYXJhMRswGQYDVQQKEwJ0dW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0Vuz21uZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAQAwdAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGQqXc
> wR4pYiO4z42/j9Ijjenh75tCKMhW51az8copP1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIzJwK5+N3x/kfa2EHU6idlavt
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/trustpoint* # commit-buffer
Firepower-chassis /security/trustpoint #
```

다음에 할 작업

Trust anchor 또는 인증 증명에서 키 링 인증서를 받아 키 링으로 가져옵니다.

## 키 링으로 인증서 가져오기

시작하기 전에

- 키 링 인증서에 대한 인증서 체인을 포함하는 신뢰 지점을 구성합니다.
- Trust anchor 또는 인증 증명에서 키 링 인증서를 가져옵니다.

절차

- 단계 1** 보안 모드를 입력합니다.  
Firepower-chassis #**scope security**
- 단계 2** 인증서를 수신할 키 링에 대한 컨피그레이션 모드로 들어갑니다.  
Firepower-chassis /security # **scopekeyringkeyring-name**
- 단계 3** 키 링 인증서를 수신한 Trust anchor 또는 인증 증명에 대한 신뢰 지점을 지정합니다.  
Firepower-chassis /security/keyring # **settrustpointname**
- 단계 4** 키 링 인증서를 입력 및 업로드할 대화 상자를 엽니다.  
Firepower-chassis /security/keyring # **setcert**

프롬프트에 Trust anchor 또는 인증 증명으로부터 받은 인증서의 텍스트를 붙여넣습니다. 인증서의 바로 다음 줄에 ENDOFBUF를 입력하여 인증서 입력을 완료합니다.

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

단계 5 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/keyring # commit-buffer
```

다음 예에서는 신뢰 지점을 지정하고 인증서를 키 링으로 가져옵니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # set trustpoint tPoint10
Firepower-chassis /security/keyring* # set cert
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
> -----BEGIN CERTIFICATE-----
> MIIB/zCCAwgCAQAwZkxCzAJBgNVBAYTAlVTMQswCQYDVQQLIEwJJDQTEVMBMGA1UE
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVBQKKEwxFeGFtcGxlIEluYy4xEzARBgNVBAS
> C1Rlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWZHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsfvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBKOND1
> GMbkPayV1QjbG4MD2dx2+H8EH3LMTdZrgKvPxPTE+bf5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzC190306Mg51zq1zXcz75+VFj2I6rH9asckC1d3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtv1WvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L4317IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 할 작업

HTTPS 서비스를 키 링으로 구성합니다.

## HTTPS 구성



주의 HTTPS에서 사용하는 포트 및 키 링 변경을 포함하여 HTTPS 컨피그레이션을 완료한 후 트랜잭션을 저장하거나 커밋하자마자 모든 현재 HTTP 및 HTTPS 세션이 종료됩니다.

절차

단계 1 시스템 모드를 입력합니다.

```
Firepower-chassis# scope system
```

단계 2 시스템 서비스 모드를 입력합니다.

```
Firepower-chassis /system # scope services
```

단계 3 HTTPS 서비스를 활성화합니다.

```
Firepower-chassis /system/services # enable https
```

단계 4 (선택 사항) HTTPS 연결에 사용할 포트를 지정합니다.

```
Firepower-chassis /system/services # set https port port-num
```

단계 5 (선택 사항) HTTPS에 대해 생성한 키 링의 이름을 지정합니다.

```
Firepower-chassis /system/services # set https keyring keyring-name
```

단계 6 (선택 사항) 도메인에서 사용하는 Cipher Suite 보안 레벨을 지정합니다.

```
Firepower-chassis /system/services # set https cipher-suite-mode cipher-suite-mode
```

*cipher-suite-mode*는 다음 키워드 중 하나일 수 있습니다.

- **high-strength**
- **medium-strength**
- **low-strength**
- **custom** - 사용자 정의 Cipher Suite 사양 문자열을 지정할 수 있습니다.

단계 7 (선택 사항) **cipher-suite-mode**가 **custom**으로 설정된 경우 도메인에 대한 Cipher Suite 보안의 커스텀 레벨을 지정합니다.

```
Firepower-chassis /system/services # set https cipher-suite cipher-suite-spec-string
```

*cipher-suite-spec-string*은 최대 256자이며 OpenSSL Cipher Suite 사양을 준수해야 합니다. 공백 또는 특수 문자를 사용할 수 없습니다. 단, !(느낌표), +(덧셈 기호), -(하이픈), :(콜론)은 사용할 수 있습니다. 자세한 내용은 [http://httpd.apache.org/docs/2.0/mod/mod\\_ssl.html#sslcipher-suite](http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslcipher-suite)를 참조하십시오.

예를 들어 FXOS에서 기본값으로 사용하는 중간 강도 사양 문자열은 다음과 같습니다.

```
ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL
```

참고 **cipher-suite-mode**가 **custom** 이외의 값으로 설정된 경우 이 옵션이 무시됩니다.

단계 8 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

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

다음 예에서는 HTTPS를 활성화하고, 포트 번호를 443으로 설정하고, 키 링 이름을 **kring7984**로 설정하고, Cipher Suite 보안 레벨을 **high**로 설정하고, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # enable https
Firepower-chassis /system/services* # set https port 443
Warning: When committed, this closes all the web sessions.
Firepower-chassis /system/services* # set https keyring kring7984
Firepower-chassis /system/services* # set https cipher-suite-mode high
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

## HTTPS 포트 변경

HTTPS 서비스는 기본적으로 포트 443에서 활성화되어 있습니다. 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 # set https port port-number
```

*port-number*에 1~65535의 정수를 지정합니다. HTTPS는 기본적으로 포트 443에서 활성화되어 있습니다.

단계 4 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

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

HTTPS 포트를 변경한 후에는 현재의 모든 HTTPS 세션이 종료됩니다. 사용자는 다음과 같이 새 포트를 사용하여 Firepower Chassis Manager에 다시 로그인해야 합니다.

```
https://<chassis_mgmt_ip_address>:<chassis_mgmt_port>
```

이때 <chassis\_mgmt\_ip\_address>는 사용자가 초기 컨피그레이션을 설정하는 동안 입력한 Firepower 새시의 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 #
```

## 키 링 삭제

### 절차

단계 1 보안 모드를 입력합니다.

```
Firepower-chassis #scope security
```

단계 2 명명된 키 링을 삭제합니다.

```
Firepower-chassis /security # deletekeyringname
```



단계 3 트랜잭션을 커밋합니다.

```
Firepower-chassis /security # commit-buffer
```

다음 예에서는 사용자 계정을 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete keyring key10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

## 신뢰 지점 삭제

시작하기 전에

신뢰 지점이 키 링에서 사용하지 않음을 확인합니다.

절차

단계 1 보안 모드로 들어갑니다.

```
Firepower-chassis# scopesecurity
```

단계 2 명명된 신뢰 지점을 삭제합니다.

```
Firepower-chassis /security # deletetrustpointname
```

단계 3 트랜잭션을 커밋합니다.

```
Firepower-chassis /security # commit-buffer
```

다음 예에서는 신뢰 지점을 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete trustpoint tPoint10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

## HTTPS 비활성화

절차

단계 1 시스템 모드를 입력합니다.

```
Firepower-chassis# scope system
```

단계 2 시스템 서비스 모드를 입력합니다.

```
Firepower-chassis /system # scope services
```

단계 3 HTTPS 서비스를 비활성화합니다.

```
Firepower-chassis /system/services # disable https
```

단계 4 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

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

다음 예에서는 HTTPS를 비활성화하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # disable https
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

## AAA 구성

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

## AAA 정보

AAA는 컴퓨터 리소스에 대한 액세스 제어를 위한 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 새시에 대한 관리 연결을 인증하도록 FXOS 새시를 구성할 수 있습니다.

- HTTPS
- SSH
- 시리얼 콘솔

### 권한 부여

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 검증 및 과금 간 상호 작용

인증을 단독으로 사용하거나 권한 부여 및 어카운팅과 함께 사용할 수 있습니다. 인증에서는 항상 먼저 사용자를 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 새시는 사용자가 사용자 프로파일을 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 제공자의 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 계정은 만료되지 않는 비밀번호를 가져야 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, 시간제한 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 LDAP 사용자의 userdn이 255자를 초과하면 사용자 로그인이 실패합니다.

다음에 할 작업

LDAP 공급업체를 만듭니다.

## 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 계정은 만료되지 않는 비밀번호를 가져야 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(Common Name)과 정확하게 일치해야 합니다. IP 주소가 지정되지 않았다면 DNS 서버를 구성해야 합니다.

- 단계 4** (선택 사항) 사용자 역할 및 로케일에 대한 값을 저장하는 LDAP 특성을 설정합니다.  
**Firepower-chassis /security/ldap/server # set attribute attr-name**  
 이 속성은 항상 이름-값의 쌍입니다. 시스템은 사용자 레코드를 쿼리하여 이 특성 이름과 일치하는 값을 찾습니다.  
 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 DN 길이는 최대 255자에서 CN=\$userid 길이를 뺀 문자 수로 설정될 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 권한이 있는 LDAP 데이터베이스 어카운트의 고유 이름(DN)을 설정합니다.  
**Firepower-chassis /security/ldap/server # set binddn binddn-name**  
 지원되는 최대 문자열 길이는 ASCII 문자 255자입니다.
- 단계 7** (선택 사항) 정의된 필터와 일치하는 사용자 이름으로 LDAP 검색을 제한합니다.  
**Firepower-chassis /security/ldap/server # set filter filter-value**  
 이 값은 기본 필터가 LDAP 제공자에 대해 설정되지 않은 경우 필요합니다.
- 단계 8** Bind DN(바인드 DN) 필드에 지정된 LDAP 데이터베이스 어카운트의 비밀번호를 지정합니다.  
**Firepower-chassis /security/ldap/server # set password**  
 공백, \$(섹션 기호), ?(물음표), =(등호)를 제외한 어떤 표준 ASCII 문자도 입력할 수 있습니다.  
 비밀번호를 설정하려면 **set password** 명령을 입력한 후 **Enter**를 누르고 프롬프트에 키 값을 입력합니다.
- 단계 9** (선택 사항) Firepower eXtensible 운영 체제에서 사용자를 인증하기 위해 이 제공자를 사용하는 순서를 지정합니다.  
**Firepower-chassis /security/ldap/server # set order order-num**
- 단계 10** (선택 사항) LDAP 서버와의 통신에 사용되는 포트를 지정합니다. 표준 포트 번호는 389입니다.  
**Firepower-chassis /security/ldap/server # set port port-num**
- 단계 11** LDAP 서버와 통신할 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.
- LDAP에서는 STARTTLS를 사용합니다. 그러면 포트 389를 사용하는 암호화 통신이 가능합니다.

단계 12 시간이 초과되기 전에 시스템이 LDAP 데이터베이스에 연결을 시도하는 데 필요한 시간(초)을 지정합니다.

```
Firepower-chassis /security/ldap/server # set timeout timeout-num
```

1~60초의 정수를 입력하거나 0(숫자 0)을 입력하여 LDAP 제공자에 지정된 전역 시간제한 값을 사용합니다. 기본값은 30초입니다.

단계 13 LDAP 제공자 또는 서버 상세정보를 제공하는 벤더를 지정합니다.

```
Firepower-chassis /security/ldap/server # set vendor {ms-ad | openldap}
```

옵션은 다음과 같습니다.

- **ms-ad** — LDAP 제공자가 Microsoft Active Directory입니다.
- **openldap** — LDAP 제공자가 Microsoft Active Directory가 아닙니다.

단계 14 (선택 사항) 인증서 해지 목록 확인을 활성화합니다.

```
Firepower-chassis /security/ldap/server # set revoke-policy {strict | relaxed}
```

참고 이 컨피그레이션은 SSL 연결이 활성화된 경우에만 적용됩니다.

단계 15 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/ldap/server # commit-buffer
```

다음의 예에서는 10.193.169.246이라는 이름의 LDAP 서버 인스턴스를 생성하고 bind, 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 bind
"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 bind, 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 bind
"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 제공자의 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 시간제한 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 **set password** 명령을 입력한 후 **Enter**를 누르고 프롬프트에 키 값을 입력합니다.

단계 6 (선택 사항) 이 서버에 시도할 순서를 지정합니다.

```
Firepower-chassis /security/radius/server # set order order-num
```

단계 7 (선택 사항) 서버가 다운되었다고 인지할 때까지 RADIUS 서버와의 통신을 재시도할 횟수를 설정합니다.

```
Firepower-chassis /security/radius/server # set retries retry-num
```



- 단계 8 서버가 다운되었다고 인지할 때까지 시스템이 RADIUS 서버의 응답을 대기해야 하는 시간 간격을 지정합니다.

```
Firepower-chassis /security/radius/server # set timeout seconds
```

팁 RADIUS 제공자에 대한 2단계 인증을 선택한 경우, 더 높은 Timeout(시간제한) 값을 구성하는 것이 좋습니다.

- 단계 9 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
Firepower-chassis /security/radius/server # commit-buffer
```

다음 예에서는 radiusserv7이라는 이름의 서버 인스턴스를 생성하고 인증 포트를 5858로 설정하고 키를 radiuskey321로 설정하고 순서를 2로 설정하고 재시도 횟수를 4로 설정하며 시간제한을 30으로 설정하고 2단계 인증을 활성화하며 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # create server radiusserv7
Firepower-chassis /security/radius/server* # set authport 5858
Firepower-chassis /security/radius/server* # set key
Enter the key: radiuskey321
Confirm the key: radiuskey321
Firepower-chassis /security/radius/server* # set order 2
Firepower-chassis /security/radius/server* # set retries 4
Firepower-chassis /security/radius/server* # set timeout 30
Firepower-chassis /security/radius/server* # commit-buffer
Firepower-chassis /security/radius/server #
```

## RADIUS 제공자 삭제

### 절차

- 단계 1 보안 모드를 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+ 제공자의 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+ 시간제한 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 만듭니다.

### 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 **set password** 명령을 입력한 후 **Enter**를 누르고 프롬프트에 키 값을 입력합니다.
- 단계 5 (선택 사항) 이 서버에 시도할 순서를 지정합니다.  
Firepower-chassis /security/tacacs/server # **set order** *order-num*
- 단계 6 서버가 다운되었다고 인지할 때까지 시스템이 TACACS+ 서버의 응답을 대기해야 하는 시간 간격을 지정합니다.  
Firepower-chassis /security/tacacs/server # **set timeout** *seconds*  
팁 TACACS+ 제공자에 대한 2단계 인증을 선택한 경우, 더 높은 Timeout(시간제한) 값을 구성하는 것이 좋습니다.
- 단계 7 (선택 사항) TACACS+ 서버와의 통신에 사용되는 포트를 지정합니다.  
Firepower-chassis /security/tacacs/server # **set port** *port-num*
- 단계 8 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
Firepower-chassis /security/tacacs/server # **commit-buffer**

다음 예에서는 tacacsserv680이라는 이름의 서버 인스턴스를 생성하고 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 콘솔로의 syslog 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 긴급도 감소 순으로 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 이상의 메시지를 표시합니다. 수준 옵션은 긴급도 감소 순으로 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 정보를 syslog 파일에 쓰는 기능을 활성화하거나 비활성화합니다.

```
Firepower-chassis /monitoring # {enable | disable} syslog file
```
- 단계 7 메시지가 로깅된 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 긴급도 감소 순으로 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 가장 오래된 메시지에 최신 메시지를 덮어쓰기 시작하기 전에 최대 파일 크기(바이트 단위)를 지정합니다. 범위는 4096~4194304바이트입니다.

```
Firepower-chassis /monitoring # set syslog file size filesize
```

**단계 10** 외부 syslog 서버 최대 3개에 syslog 메시지를 전송하도록 구성합니다.

a) 외부 syslog 서버 최대 3개에 syslog 메시지 전송하는 기능을 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 긴급도 감소 순으로 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. 활성화하거나 비활성화하려는 각 로컬 소스에 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 호스트의 IP 주소를 확인해야 하는 경우 DNS 서버를 지정해야 합니다. 예를 들어 DNS 서버를 구성하지 않으면 Firepower 새시에서 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 경우 임의의 순서로만 서버를 검색합니다. 로컬 관리 명령에서 DNS 서버 조회가 필요할 경우 임의의 순서대로 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 다음과 같이 합니다.

```
Firepower-chassis /system/services # create dns {ip-addr | ip6-addr}
```

- 지정된 IPv4 또는 IPv6 주소가 있는 DNS 서버를 삭제하려면 다음과 같이 합니다.

```
Firepower-chassis /system/services # delete dns {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 #
```







## 인터페이스 관리

- [Firepower Security Appliance 인터페이스 정보, 143 페이지](#)
- [인터페이스 속성 편집, 145 페이지](#)
- [포트 채널 생성, 146 페이지](#)
- [분할 케이블 구성, 147 페이지](#)

### Firepower Security Appliance 인터페이스 정보

FXOS 새시는 단일 인터페이스 뿐만 아니라 EtherChannel(port-channel) 인터페이스도 지원합니다. EtherChannel 인터페이스는 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

### 인터페이스 유형

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 관리 인터페이스 1개만 할당할 수 있습니다.

Firepower Threat Defense 애플리케이션에서 물리적 관리 인터페이스는 논리적 진단 인터페이스와 논리적 관리 인터페이스 간에 공유됩니다. 논리적 관리 인터페이스는 디바이스에 있는 다른 인터페이스와 분리되어 있습니다. 이 인터페이스는 디바이스를 Firepower Management Center에 설치하고 등록하는 데 사용됩니다. 또한 별도의 SSH 서버를 실행하고 자체 로컬 인증, IP 주소 및 정적 라우팅을 사용합니다. CLI에서 **configure network** 명령을 사용하여 설정을 구성하고 **Management CenterDevices(디바이스) > Device Management(디바이스 관리) > Devices(디바이스) > Management(관리)** 영역에서 IP 주소를 변경할 수 있습니다.

논리적 진단 인터페이스는 **Management CenterDevices(디바이스) > Device Management(디바이스 관리) > Interfaces(인터페이스)** 화면에서 나머지 데이터 인터페이스와 함께 구성할 수 있습니다. 진단 인터페이스 사용은 선택 사항입니다. 진단 인터페이스 및 데이터 인터페이스는 LDAP

또는 RADIUS 외부 인증을 허용합니다. 데이터 인터페이스에서 SSH 액세스를 허용하지 않으려면, 예를 들어 SSH 액세스를 지원하는 진단 인터페이스를 구성하도록 선택할 수 있습니다. 진단 인터페이스는 관리 트래픽만 허용하며 통과 트래픽은 허용하지 않습니다.

- **Firepower-eventing** -- Firepower Threat Defense 디바이스의 보조 관리 인터페이스입니다. 이 인터페이스를 사용하려면 Firepower Threat Defense CLI에서 해당 IP 주소 및 기타 매개변수를 구성해야 합니다. 예를 들면 관리 트래픽을 이벤트(예: 웹 이벤트)에서 분리할 수 있습니다. Firepower Management Center 명령 참조에서 **configure network** 명령을 참조하십시오.
- **클러스터** -- 클러스터된 논리적 디바이스에 사용되는 특수 인터페이스 유형입니다. 이 유형은 유닛 클러스터 간 통신을 지원하는 클러스터 제어 링크에 자동으로 할당됩니다. 기본적으로, 클러스터 제어 링크는 Port-channel 48에서 자동으로 생성됩니다.



참고

Firepower Management Center 또는 Firepower Threat Defense CLI를 사용하여 두 개의 업링크, Breakout 또는 데이터 포트 인터페이스를 인라인 쌍으로 구성할 수 있습니다. 두 포트가 인라인 쌍으로 구성되면 단일 인터페이스로 작동합니다. 그러면 이 컨피그레이션이 FXOS 새시로 전파됩니다.

인라인 쌍에 대한 다음 제약 조건을 참조하십시오.

- 두 개의 포트 인터페이스는 고유해야 합니다. 한 인라인 쌍에 추가한 포트는 다른 인라인 쌍에 추가할 수 없습니다.
- 업링크 포트, 데이터 포트 또는 Breakout 포트만 인라인 쌍으로 구성할 수 있습니다.

자세한 내용은 Firepower Management Center 환경 설정 가이드의 IPS 전용 인터페이스의 인라인 설정 구성 항목을 참조하십시오.

## 하드웨어 바이패스 쌍

Firepower Threat Defense의 경우 Firepower 9300 및 4100 Series의 특정 인터페이스 모듈에서 하드웨어 바이패스 기능을 활성화할 수 있습니다. 하드웨어 바이패스는 정전 중에도 트래픽이 인라인 인터페이스 쌍 사이에서 계속 흐르도록 합니다. 이 기능은 소프트웨어 또는 하드웨어 오류의 경우 네트워크 연결성을 유지 관리하는 데 사용될 수 있습니다.

하드웨어 바이패스 기능은 하드웨어 바이패스 애플리케이션 내에서 구성됩니다. 이러한 인터페이스를 하드웨어 바이패스 쌍으로 사용할 필요가 없습니다. 이들은 ASA 및 Firepower Threat Defense 애플리케이션에서 모두 일반 인터페이스로 사용할 수 있습니다. Breakout 포트에 대해 하드웨어 바이패스 지원 인터페이스를 구성할 수 없습니다. 하드웨어 바이패스 기능을 사용하려면 포트를 EtherChannel로 구성하지 마십시오. 그렇게 하지 않으면 이러한 인터페이스를 일반 인터페이스 모드에서 EtherChannel 멤버로 포함할 수 있습니다.

Firepower Threat Defense는 다음 모델에서 특정 네트워크 모듈의 인터페이스 쌍에 대해 하드웨어 바이패스를 지원합니다.

- Firepower 9300
- Firepower 4100 Series

이러한 모델에 대해 지원되는 하드웨어 바이패스 네트워크 모듈은 다음과 같습니다.

- Firepower 6 포트 1G SX FTW Network Module single-wide(FPR-NM-6X1SX-F)
- Firepower 6 포트 10G SR FTW Network Module single-wide(FPR-NM-6X10SR-F)
- Firepower 6 포트 10G LR FTW Network Module single-wide(FPR-NM-6X10LR-F)
- Firepower 2 포트 40G SR FTW Network Module single-wide(FPR-NM-2X40G-F)

하드웨어 바이패스는 다음 포트 쌍만 사용할 수 있습니다.

- 1 및 2
- 3 및 4
- 5 및 6

## Jumbo Frame Support

FXOS 새시에서는 기본적으로 점보 프레임 지원이 활성화되어 있습니다. FXOS 새시에 설치된 특정 논리적 디바이스에서 점보 프레임 지원을 활성화하려면 논리적 디바이스에서 인터페이스에 대한 적절한 MTU 설정을 구성해야 합니다.

FXOS 새시의 애플리케이션에 대해 지원되는 최대 MTU는 9184입니다.

## 인터페이스 속성 편집

절차

단계 1 인터페이스 모드를 입력합니다.

```
scope eth-uplink
scope fabric a
```

단계 2 인터페이스를 활성화합니다.

```
enter interface interface_id
활성화
```

예제:

```
Firepower /eth-uplink/fabric # enter interface Ethernet1/8
Firepower /eth-uplink/fabric/interface # enable
```

참고 이미 포트 채널의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. 포트 채널의 멤버인 인터페이스에서 **enter interface** 또는 **scope interface** 명령을 사용하는 경우 객체가 존재하지 않음을 알리는 오류가 표시됩니다. 포트 채널에 인터페이스를 추가하기 전에 **enter interface** 명령을 사용하여 인터페이스를 수정해야 합니다.

단계 3 (선택 사항) 인터페이스 유형을 설정합니다.

```
setport-type {data | firepower-eventing | mgmt | cluster}
```

예제:

```
Firepower /eth-uplink/fabric/interface # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. **cluster** 키워드를 선택하지 마십시오.

단계 4 (선택 사항) 인터페이스 속도를 설정합니다.

```
setadmin-speed {10gbps | 1gbps}
```

예제:

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

단계 5 컨피그레이션을 커밋합니다.

```
commit-buffer
```

## 포트 채널 생성

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

FXOS 새시에서 EtherChannel을 만들면, 물리적 링크가 가동 중이더라도 EtherChannel은 물리적 디바이스에 할당될 때까지 **Suspended**(일시 중단) 상태로 유지됩니다. 다음의 상황에서는 EtherChannel의 **Suspended**(일시 중단) 상태가 해제됩니다.

- EtherChannel이 독립형 논리적 디바이스에 대한 데이터 포트 또는 관리 포트에 추가됩니다.
- EtherChannel이 클러스터의 일부인 논리적 디바이스에 대한 관리 포트 또는 CCL 포트에 추가됩니다.
- EtherChannel이 클러스터의 일부인 논리적 디바이스에 대한 데이터 포트에 추가되며 보안 모듈 하나 이상이 클러스터에 조인됩니다.

EtherChannel은 논리적 디바이스에 할당될 때까지 나타나지 않습니다. EtherChannel을 논리적 디바이스에서 제거하거나 논리적 디바이스가 삭제된 경우, EtherChannel은 **Suspended** 상태로 전환됩니다.

시작하기 전에

FXOS 새시에서는 활성 LACP(Link Aggregation Control Protocol) 모드에서 EtherChannel만 지원합니다. 최고의 호환성을 위해 연결 스위치 포트를 Active(활성) 모드로 설정하는 것이 좋습니다.

절차

단계 1 인터페이스 모드를 입력합니다.

```
scope eth-uplink
scope fabric a
```

단계 2 포트 채널을 생성합니다.

**create port-channel *id***

활성화

단계 3 멤버 인터페이스를 할당합니다.

**create member-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.

**set port-type {data | mgmt | cluster}**

예제:

```
Firepower /eth-uplink/fabric/port-channel # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

단계 5 (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.

**set speed {10gbps | 1gbps}**

예제:

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

단계 6 컨피그레이션을 커밋합니다.

**commit-buffer**

## 분할 케이블 구성

다음 절차에서는 FXOS 새시에서 사용할 분할 케이블을 구성하는 방법을 보여줍니다. 분할 케이블을 사용하여 40Gbps 포트 1개 대신 10Gbps 포트 4개를 제공할 수 있습니다.

시작하기 전에

Breakout 포트에 대해 하드웨어 바이패스 지원 인터페이스를 구성할 수 없습니다.

## 절차

단계 1 다음 명령을 사용하여 새 분할 케이블을 생성합니다.

a) 케이블 모드를 입력합니다.

```
scope cabling  
scope fabric a
```

b) 분할 케이블을 생성합니다.

```
create breakout network_module_slot port
```

예제:

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

c) 컨피그레이션을 커밋합니다.

```
commit-buffer
```

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

단계 2 다음 명령을 사용하여 분할 포트를 활성화하고 구성합니다.

a) 인터페이스 모드를 입력합니다.

```
scope eth-uplink
```

```
scope fabric a
```

```
scope aggr-interface network_module_slot port
```

참고 이미 포트 채널의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. 포트 채널의 멤버인 인터페이스에서 **enter interface** 또는 **scope interface** 명령을 사용하는 경우 객체가 존재하지 않음을 알리는 오류가 표시됩니다. 포트 채널에 인터페이스를 추가하기 전에 **enter interface** 명령을 사용하여 인터페이스를 수정해야 합니다.

b) **set** 명령을 사용하여 인터페이스 속도 및 포트 유형을 구성합니다.

```
enable 또는 disable 명령을 사용하여 인터페이스의 관리 상태를 설정합니다.
```

c) 컨피그레이션을 커밋합니다.

```
commit-buffer
```



## 논리적 디바이스

- 논리적 디바이스 정보, 149 페이지
- 독립형 ASA 논리적 디바이스 생성, 150 페이지
- 독립형 위협 방어 논리적 디바이스 생성, 152 페이지
- 클러스터 구축, 156 페이지
- 서비스 체이닝 정보, 179 페이지
- 독립형 논리적 디바이스에 Radware DefensePro 서비스 체인 구성, 182 페이지
- 내장 새시 클러스터에 Radware DefensePro 서비스 체인 구성, 183 페이지
- UDP/TCP 포트 열기 및 vDP 웹 서비스 활성화, 186 페이지
- 애플리케이션 콘솔 또는 데코레이터에 연결, 187 페이지
- 논리적 디바이스 삭제, 188 페이지
- 논리적 디바이스와 연결되지 않은 애플리케이션 인스턴스 삭제, 189 페이지

## 논리적 디바이스 정보

논리적 디바이스를 생성할 때 FXOS 새시 관리자(Supervisor)는 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 지정된 보안 모듈/엔진에 푸시하여 논리적 디바이스를 구축합니다. 또는, 인트라 새시(intra-chassis) 클러스터의 경우, Firepower 새시에 설치된 모든 보안 모듈에 적용합니다.

다음 2가지 유형의 논리적 디바이스 중 하나를 생성할 수 있습니다.



참고 여러 보안 모듈을 지원하는 FXOS 새시에서 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.



참고 독립형 논리적 디바이스를 구성하는 경우, 새시에 있는 모든 모듈에 동일한 소프트웨어 유형을 설치해야 하며 다른 소프트웨어 유형은 현재 지원되지 않습니다. 참고로, 모듈은 특정 디바이스 유형의 다른 버전을 실행할 수 있지만 모든 모듈은 동일한 유형의 논리적 디바이스로 구성되어야 합니다.

- 독립형 - Firepower 새시에 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 9300과 같은 다중 모듈 디바이스는 인트라 새시 클러스터링(intra-chassis clustering)을 지원합니다.

## 독립형 ASA 논리적 디바이스 생성

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



참고 선택적으로, 서드파티 Radware DefensePro 가상 플랫폼을 보안 모듈의 ASA 방화벽 앞에 DDoS 탐지 및 완화 서비스로 설치할 수 있습니다([서비스 체이닝 정보](#), 179 페이지 참조).



참고 하나의 새시 내의 모든 모듈에 동일한 소프트웨어 유형을 설치해야 합니다. 다른 소프트웨어 유형은 지원되지 않습니다. 모듈은 특정 디바이스 유형의 다른 버전을 실행할 수 있지만 모든 모듈은 동일한 유형의 논리적 디바이스로 구성되어야 합니다.

시작하기 전에

- 논리적 디바이스에 사용할 보안 모듈/엔진에 논리적 디바이스가 이미 구성되어 있는 경우 기존의 논리적 디바이스를 먼저 삭제해야 합니다([논리적 디바이스 삭제](#), 188 페이지 참조).
- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드([Cisco.com에서 이미지 다운로드](#), 48 페이지 참조)한 다음 해당 이미지를 FXOS 새시에 다운로드합니다([FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드](#), 50 페이지 참조).



- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다.

## 절차

- 
- 단계 1** 보안 서비스 모드를 입력합니다.  
Firepower# **scope ssa**
- 단계 2** 논리적 디바이스를 생성합니다.  
Firepower /ssa # **create logical-device** *device\_name* **asa** *slot\_id* **standalone**
- 단계 3** 논리적 디바이스에 대한 설명을 입력합니다.  
Firepower /ssa/logical-device\* # **set description** "*logical device description*"
- 단계 4** 논리적 디바이스에 관리 및 데이터 인터페이스를 할당합니다.  
Firepower /ssa/logical-device\* # **create external-port-link** *name* *interface\_name* **asa**  
Firepower-chassis /ssa/logical-device/external-port-link\* # **exit**
- 단계 5** 관리 부트스트랩 정보를 구성합니다.
- 부트스트랩 객체를 생성합니다.  
Firepower /ssa/logical-device\* # **create mgmt-bootstrap** **asa**
  - 비밀번호 사용을 생성합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap\* # **create bootstrap-key-secret** **PASSWORD**
  - 비밀번호 값을 설정합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret\* # **set value**  
값: *password*
  - 비밀번호 컨피그레이션 범위를 종료합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret\* # **exit**
  - 관리 IP 주소를 구성합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap\* # **create ipv4** *slot\_id* **default**
  - 게이트웨이 주소를 설정합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4\* # **set gateway** *gateway\_address*
  - IP 주소 및 마스크를 설정합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4\* # **set ip** *ip\_address* **mask** *network\_mask*
  - 관리 IP 컨피그레이션 범위를 종료합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4\* # **exit**
  - 관리 부트스트랩 컨피그레이션 범위를 종료합니다.  
Firepower /ssa/logical-device/mgmt-bootstrap\* # **exit**
- 단계 6** 컨피그레이션을 커밋합니다.  
**commit-buffer**  
시스템 컨피그레이션에 트랜잭션을 커밋합니다.
-

예

```

Firepower# scope ssa
Firepower /ssa # create logical-device MyDevice1 asa 1 standalone
Firepower /ssa/logical-device* # set description "logical device description"
Firepower /ssa/logical-device* # create external-port-link inside Ethernet1/1 asa
Firepower /ssa/logical-device/external-port-link* # set description "inside link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link management Ethernet1/7 asa
Firepower /ssa/logical-device/external-port-link* # set description "management link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link outside Ethernet1/2 asa
Firepower /ssa/logical-device/external-port-link* # set description "external link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value: <password>
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 default
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 1.1.1.254
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 1.1.1.1 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* # show configuration pending
+enter logical-device MyDevice1 asa 1 standalone
+  enter external-port-link inside Ethernet1/1 asa
+    set decorator ""
+    set description "inside link"
+  exit
+  enter external-port-link management Ethernet1/7 asa
+    set decorator ""
+    set description "management link"
+  exit
+  enter external-port-link outside Ethernet1/2 asa
+    set decorator ""
+    set description "external link"
+  exit
+  enter mgmt-bootstrap asa
+    enter bootstrap-key-secret PASSWORD
+      set value
+    exit
+    enter ipv4 1 default
+      set gateway 1.1.1.254
+      set ip 1.1.1.1 mask 255.255.255.0
+    exit
+  exit
+  set description "logical device description"
+exit
Firepower /ssa/logical-device* # commit-buffer

```

## 독립형 위협 방어 논리적 디바이스 생성

FXOS 새시에 설치된 각각의 보안 모듈/엔진에 독립형 논리적 디바이스를 생성할 수 있습니다. Firepower 9300과 같은 다중 모듈 디바이스에서는 클러스터가 구성되어 있는 경우 독립형 논리적 디바이스를 생성할 수 없습니다. 독립형 디바이스를 구성하려면 먼저 클러스터를 삭제해야 합니다.



참고 선택적으로, 서드파티 Radware DefensePro 가상 플랫폼을 보안 모듈의 Firepower Threat Defense 논리적 디바이스 앞에 DDoS 탐지 및 완화 서비스로 설치할 수 있습니다([서비스 체이닝 정보](#), 179 페이지 참조).



참고 하나의 새시 내의 모든 모듈에 동일한 소프트웨어 유형을 설치해야 합니다. 다른 소프트웨어 유형은 지원되지 않습니다. 모듈은 특정 디바이스 유형의 다른 버전을 실행할 수 있지만 모든 모듈은 동일한 유형의 논리적 디바이스로 구성되어야 합니다.

#### 시작하기 전에

- 논리적 디바이스에 사용할 보안 모듈/엔진에 논리적 디바이스가 이미 구성되어 있는 경우, 기존의 논리적 디바이스를 먼저 삭제해야 합니다([논리적 디바이스 삭제](#), 188 페이지 참고).
- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드([Cisco.com에서 이미지 다운로드](#), 48 페이지 참고)한 다음 해당 이미지를 FXOS 새시에 다운로드합니다([FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드](#), 50 페이지 참고).
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 최소 하나 이상의 데이터 유형 인터페이스도 구성해야 합니다. 또는 Firepower 이벤트 처리 인터페이스를 생성하여 모든 이벤트 트래픽을 전달할 수 있습니다(예: 웹 이벤트).

#### 절차

단계 1 보안 서비스 모드를 입력합니다.

```
Firepower# scope ssa
```

단계 2 논리적 디바이스를 생성합니다.

```
Firepower /ssa # createlogical-device device_name ftdslot_id standalone
```

*device\_name*은 FXOS 새시 관리자(Supervisor)가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 디바이스 이름이 아닙니다.

단계 3 (선택 사항) Radware DefensePro 이미지를 설치합니다.

```
Firepower /ssa # scopeslot_id
```

```
Firepower/ssa/slot_id # createapp-instance vdp
```

논리적 디바이스 컨피그레이션을 커밋하기 위한 이 절차의 마지막 단계를 수행한 후에는 Firepower Threat Defense 논리적 디바이스로 서비스 체인에서 Radware DefensePro 테코레이터를 계속 구성해야 합니다. [독립형 논리적 디바이스에 Radware DefensePro 서비스 체인 구성](#), 182 페이지 절차를 4단계부터 참조하십시오.

단계 4 논리적 디바이스에 관리 및 데이터 인터페이스를 할당합니다.

```
Firepower /ssa/logical-device* # createexternal-port-linknameinterface_name ftd
```

```
Firepower-chassis /ssa/logical-device/external-port-link* # exit
```

단계 5 관리 부트스트랩 매개변수를 구성합니다.

a) 부트스트랩 객체를 생성합니다.

```
Firepower /ssa/logical-device* # createmgmt-bootstrap ftd
```

b) Firepower Management Center 관리에 사용할 IP 주소를 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-key FIREPOWER_MANAGER_IP
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value IP_address
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
```

- c) 논리적 디바이스가 작동할 모드(Routed(라우팅됨) 또는 Transparent(투명))를 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-keyFIREWALL_MODE
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value firewall_mode
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
```

- d) 디바이스와 Firepower Management Center 간에 공유할 키를 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-key-secretREGISTRATION_KEY
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
```

값: *registration\_key*

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
```

- e) 논리적 디바이스에 사용할 비밀번호를 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-key-secretPASSWORD
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
```

값: *password*

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
```

- f) 논리적 디바이스의 정규화된 호스트 이름을 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-keyFQDN
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value fqdn
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
```

- g) 논리적 디바이스에서 사용할 쉼표로 구분된 DNS 서버 목록을 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-keyDNS_SERVERS
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value dns_servers
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
```

- h) 논리적 디바이스의 서버 도메인을 쉼표로 구분하여 지정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createbootstrap-keySEARCH_DOMAINS
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value search_domains
```

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
```

- i) 관리 인터페이스를 구성합니다.

이것은 모든 이벤트 트래픽(예: 웹 이벤트)을 전달하는 Firepower 이벤트 트래픽 채널에 사용되는 인터페이스입니다.

다음과 같이 IPv4 관리 인터페이스 객체를 생성합니다.

- 1 관리 인터페이스 객체를 생성합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createipv4slot_idfirepower
```

- 2 게이트웨이 주소를 설정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway gateway_address
```

- 3 IP 주소 및 마스크를 설정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip ip_address mask network_mask
```

- 4 관리 IP 컨피그레이션 범위를 종료합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
```

다음과 같이 IPv6 관리 인터페이스 객체를 생성합니다.

- 1 관리 인터페이스 객체를 생성합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* # createipv6slot_idfirepower
```

- 2 게이트웨이 주소를 설정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # setgatewaygateway_address
```

- 3 IP 주소 및 접두사를 설정합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # set ipip_addressprefix-lengthprefix
```

- 4 관리 IP 컨피그레이션 범위를 종료합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # exit
```

- j) 관리 부트스트랩 모드를 종료합니다.

```
Firepower /ssa/logical-device/mgmt-bootstrap* #exit
```

단계 6 최종 사용자 라이선스 계약에 동의합니다.

- a) Firepower /ssa/logical-device/\* # exit
- b) Firepower /ssa #scope appapplication\_typeapplication\_version
- c) Firepower /ssa/app #show license-agreement
- d) Firepower /ssa/app #accept-license-agreement

단계 7 컨피그레이션을 커밋합니다.

```
commit-buffer
```

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa #create logical-device MyDevice1 ftd 1 standalone
Firepower /ssa/logical-device* # create external-port-link inside Ethernet1/1 ftd
Firepower /ssa/logical-device/external-port-link* # set description "inside link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link management Ethernet1/7 ftd
Firepower /ssa/logical-device/external-port-link* # set description "management link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create external-port-link outside Ethernet1/2 ftd
Firepower /ssa/logical-device/external-port-link* # set description "external link"
Firepower /ssa/logical-device/external-port-link* # exit
Firepower /ssa/logical-device* # create mgmt-bootstrap ftd
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key FIREPOWER_MANAGER_IP
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 10.0.0.100
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key FIREWALL MODE
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value routed
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret REGISTRATION_KEY
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value:
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
```

```

Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value:
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.31 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key FQDN
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value ftd.cisco.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key DNS_SERVERS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 192.168.1.1
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key SEARCH_DOMAINS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value search.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* # exit
Firepower /ssa # scope app ftd 6.0.0.837
Firepower /ssa/app # accept-license-agreement
Firepower /ssa/app* # commit-buffer

```

## 클러스터 구축

클러스터링을 사용하면 여러 개의 디바이스를 하나의 논리적 디바이스로 그룹화할 수 있습니다. 클러스터는 처리량 증대 및 여러 디바이스의 이중화라는 목표를 달성하는 동시에 단일 디바이스(관리, 네트워크에 통합)의 모든 편의성을 제공합니다. 여러 모듈을 포함하는 Firepower 9300은 단일 새시의 모든 모듈을 하나의 클러스터로 그룹화하는 인트라 새시 클러스터링(intra-chassis clustering)을 지원합니다. 여러 새시가 그룹화되는 새시 간 클러스터링을 사용할 수도 있습니다. Firepower 4100 Series 같은 단일 모듈 디바이스에는 새시 간 클러스터링이 유일한 옵션입니다.

## FXOS 새시의 클러스터링 정보

클러스터는 단일 논리적 유닛으로 작동하는 여러 개의 디바이스로 구성됩니다. FXOS 새시에서 클러스터를 구축하려면 다음 작업을 수행합니다.

- 유닛 간 통신에 사용되는 클러스터 제어 링크(기본값: port-channel 48)를 생성합니다. 인트라 새시 클러스터링(intra-chassis clustering)(Firepower 9300 전용)의 경우, 이 링크는 클러스터 통신에 Firepower 9300 백플레인을 활용합니다. 새시 간 클러스터링의 경우, 새시 간의 통신을 위해 물리적 인터페이스를 이 EtherChannel에 수동으로 할당해야 합니다.
- 애플리케이션 내부에 클러스터 부트스트랩 컨피그레이션을 생성합니다.

클러스터를 구축할 때, FXOS 새시 관리 프로그램은 클러스터 이름, 클러스터 제어 링크 인터페이스 및 기타 클러스터 설정을 포함하는 각 유닛에 최소한의 부트스트랩 컨피그레이션을 푸시합니다. 클러스터링 환경을 사용자 정의하려는 경우, 사용자가 일부 부트스트랩 컨피그레이션을 애플리케이션 내부에 구성할 수 있습니다.

- 데이터 인터페이스를 *Spanned* 인터페이스로 클러스터에 할당합니다.

인트라 새시 클러스터링(intra-chassis clustering)의 경우, *Spanned* 인터페이스는 새시 간 클러스터링과 마찬가지로 EtherChannel에 국한되지 않습니다. Firepower 9300 관리자(Supervisor)는 EtherChannel 기술을 내부에 사용하여 트래픽을 공유 인터페이스의 다중 모듈에 로드 밸런싱하

므로 모든 데이터 인터페이스 유형이 Spanned 모드에서 작동합니다. 새시 간 클러스터링의 경우, 모든 데이터 인터페이스에 Spanned EtherChannel을 사용해야 합니다.



참고 개별 인터페이스는 관리 인터페이스를 제외하고 지원되지 않습니다.

- 관리 인터페이스를 클러스터의 모든 유닛에 할당합니다.

다음 섹션에서는 클러스터링 개념 및 구현에 대한 자세한 정보를 제공합니다.

## 기본 유닛 및 보조 유닛 역할

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

기본 유닛에서만 모든 컨피그레이션을 수행해야 하며 이후에 컨피그레이션은 보조 유닛에 복제됩니다.

## 클러스터 제어 링크

클러스터 제어 링크는 Port-channel 48 인터페이스를 사용하여 자동으로 생성됩니다. 인트라 새시 클러스터링(intra-chassis clustering)의 경우, 이 인터페이스에는 멤버 인터페이스가 없습니다. 새시 간 클러스터링의 경우, EtherChannel에 인터페이스를 하나 이상 추가해야 합니다. 이 클러스터 유형 EtherChannel은 인트라 새시 클러스터링(intra-chassis clustering)을 위한 클러스터 통신에 Firepower 9300 백플레인을 활용합니다.

2-멤버 새시 간 클러스터의 경우 클러스터 제어 링크를 한 새시에서 다른 새시로 직접 연결하지 마십시오. 인터페이스에 직접 연결할 경우, 유닛 하나에 오류가 발생하면 클러스터 제어 링크에도 오류가 발생하므로 나머지 정상 유닛에도 오류가 발생합니다. 스위치를 통해 클러스터 제어 링크를 연결할 경우 클러스터 제어 링크는 가동 상태를 유지하여 정상 유닛을 지원합니다.

클러스터 제어 링크 트래픽에는 제어 및 데이터 트래픽이 모두 포함됩니다.

## 새시 간 클러스터링을 위한 클러스터 제어 링크 크기 조정

가능한 경우, 각 새시의 예상 처리량에 맞게 클러스터 제어 링크의 크기를 조정하여 클러스터 제어 링크가 최악의 시나리오를 처리할 수 있게 해야 합니다.

클러스터 제어 링크 트래픽은 주로 상태 업데이트 및 전달된 패킷으로 구성되어 있습니다. 클러스터 제어 링크의 트래픽 양은 언제든지 달라질 수 있습니다. 전달된 트래픽의 양은 로드 밸런싱 효율성 또는 중앙 집중식 기능에 많은 트래픽이 있는지에 따라 좌우됩니다. 예를 들면 다음과 같습니다.

- NAT의 경우 연결의 로드 밸런싱이 저하되며, 모든 반환 트래픽을 올바른 유닛으로 다시 밸런싱해야 합니다.
- 네트워크 액세스용 AAA는 중앙 집중식 기능이므로 모든 트래픽이 기본 유닛으로 전달됩니다.

- 멤버가 변경된 경우, 클러스터에서는 다량의 연결을 다시 밸런싱해야 하므로 일시적으로 많은 양의 클러스터 제어 링크 대역폭을 사용합니다.

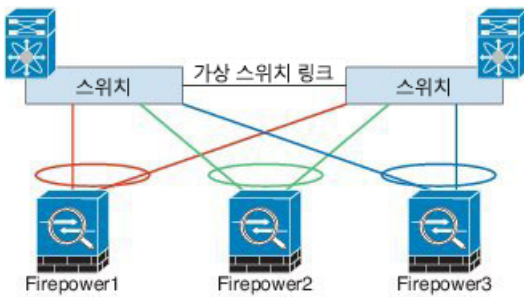
대역폭이 높은 클러스터 제어 링크를 사용하면 멤버가 변경될 경우 클러스터를 더 빠르게 통합할 수 있고 처리량 병목 현상을 방지할 수 있습니다.



참고 클러스터에 비대칭(다시 밸런싱된) 트래픽이 많은 경우 클러스터 제어 링크 크기를 늘려야 합니다.

### 새시 간 클러스터링을 위한 클러스터 제어 링크 이중화

다음 다이어그램에는 VSS(Virtual Switching System) 또는 vPC(Virtual Port Channel) 환경에서 EtherChannel을 클러스터 제어 링크로 사용하는 방법이 나와 있습니다. EtherChannel의 모든 링크가 활성화되어 있습니다. 스위치가 VSS 또는 vPC의 일부일 경우 동일한 EtherChannel 내에서 FXOS 새시 인터페이스를 연결하여 VSS 또는 vPC의 스위치와 별도로 분리할 수 있습니다. 이러한 별도의 스위치는 단일 스위치 역할을 수행하므로, 스위치 인터페이스는 동일한 EtherChannel 포트 채널 인터페이스의 멤버입니다. 이러한 EtherChannel은 디바이스 로컬이 아닌 Spanned EtherChannel입니다.



### 새시 간 클러스터링을 위한 클러스터 제어 링크 안정성

클러스터 제어 링크 기능을 보장하려면 유닛 간의 RTT(round-trip time)가 20ms 이하여야 합니다. 이러한 최대 레이턴시는 서로 다른 지리적 사이트에 설치된 클러스터 멤버와의 호환성을 개선하는 역할을 합니다. 레이턴시를 확인하려면 유닛 간의 클러스터 제어 링크에서 Ping을 수행합니다.

클러스터 제어 링크는 오류가 나거나 폐기된 패킷이 없는 안정적인 상태여야 합니다. 예를 들어, 사이트 간 구축의 경우 전용 링크를 사용해야 합니다.

### 관리 네트워크

모든 유닛을 단일 관리 네트워크에 연결하는 것이 좋습니다. 이러한 네트워크는 클러스터 제어 링크와 분리되어 있습니다.



## 관리 인터페이스

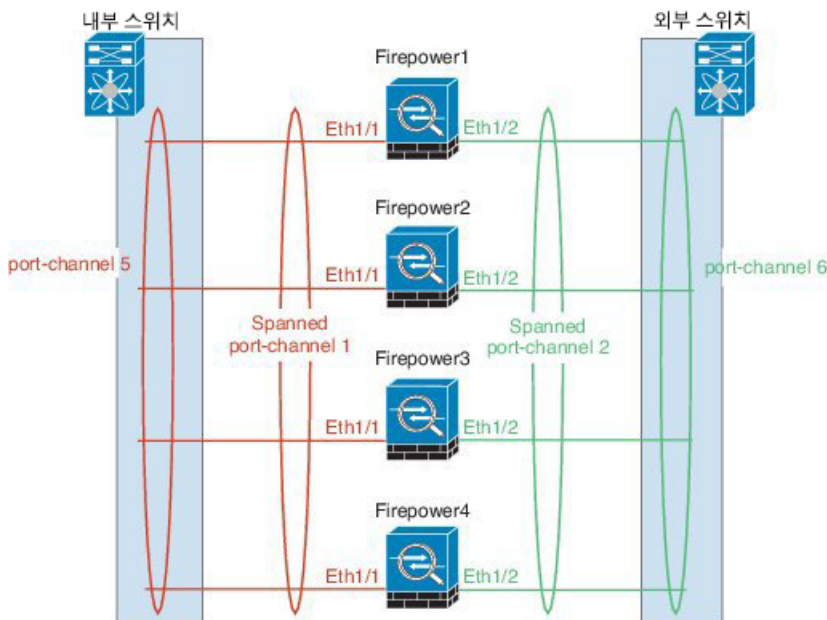
클러스터에 관리 유형 인터페이스를 할당할 수 있습니다. 이 인터페이스는 **Spanned** 인터페이스와는 다른 특수 개별 인터페이스입니다. 관리 인터페이스를 사용하면 각 유닛에 직접 연결할 수 있습니다.

ASA의 경우, 기본 클러스터 IP 주소는 현재 기본 유닛에 항상 속해 있는 클러스터를 위한 고정 주소입니다. 주소의 범위를 구성하여 현재 기본 유닛을 비롯한 각 유닛에서 해당 범위의 로컬 주소를 사용할 수 있도록 합니다. 기본 클러스터 IP 주소에서는 주소에 대한 일관된 관리 액세스를 제공합니다. 기본 유닛이 변경될 경우 기본 클러스터 IP 주소는 새 기본 유닛으로 이동되므로 클러스터는 지속적으로 원활하게 관리됩니다. 로컬 IP 주소는 라우팅에 사용되며 문제 해결에도 도움이 됩니다. 예를 들어, 현재 기본 유닛에 항상 연결되어 있는 기본 클러스터 IP 주소에 연결하여 클러스터를 관리할 수 있습니다. 로컬 IP 주소에 연결하여 개별 멤버를 관리할 수 있습니다. TFTP 또는 syslog 같은 아웃바운드 관리 트래픽의 경우 기본 유닛을 비롯한 각 유닛에서 로컬 IP 주소를 사용하여 서버에 연결합니다.

Firepower Threat Defense의 경우, 동일한 네트워크의 각 유닛에 관리 IP 주소를 할당합니다. 각 유닛을 Management Center에 추가할 때 이 IP 주소를 사용합니다.

## Spanned EtherChannel

새시당 하나 이상의 인터페이스를 클러스터 내의 모든 새시를 포괄하는 EtherChannel로 그룹화할 수 있습니다. EtherChannel에서는 채널에서 사용 가능한 모든 활성 인터페이스 전반의 트래픽을 취합합니다. Spanned EtherChannel은 라우팅 및 투명 방화벽 모드에서 모두 구성할 수 있습니다. 라우팅 모드인 경우 EtherChannel은 단일 IP 주소를 통해 라우팅된 인터페이스로 구성됩니다. 투명 모드의 경우 브리지 그룹 멤버 인터페이스가 아닌 BVI에 IP 주소가 할당됩니다. EtherChannel은 기본적인 작동 시 로드 밸런싱을 함께 제공합니다.



## 사이트 사이트별 MAC 및 IP 주소

각 클러스터 새시를 별도의 사이트 ID에 속하도록 구성할 수 있습니다.

사이트 ID는 사이트별 MAC 주소 및 IP 주소와 작동합니다. 클러스터에서 온 패킷은 사이트별 MAC 주소 및 IP 주소를 사용하는 반면, 클러스터가 수신한 패킷은 전역 MAC 주소 및 IP 주소를 사용합니다. 이 기능은 스위치가 서로 다른 두 포트의 두 사이트로부터 동일한 전역 MAC 주소를 학습하지 못하게 하는 한편, MAC 플래핑(flapping)을 일으킵니다. 대신 스위치는 사이트 MAC 주소만 학습합니다. 사이트별 MAC 주소 및 IP 주소는 Spanned EtherChannel만을 사용하는 라우팅 모드에서 지원됩니다.

사이트 ID는 LISP 검사를 사용하여 플로우 모빌리티를 활성화하는 데에도 사용됩니다.

## 클러스터링의 사전 요구 사항

새시 간 하드웨어 및 소프트웨어 요구 사항

클러스터의 모든 새시:

- Firepower 4100 Series의 경우: 모든 새시가 동일한 모델이어야 합니다. Firepower 9300의 경우: 모든 보안 모듈이 동일한 유형이어야 합니다. 새시에 있는 모든 모듈은 클러스터에 속해야 하지만 각 새시에 설치된 보안 모듈의 수는 다를 수 있습니다.
- 이미지 업그레이드 시 동일한 FXOS 소프트웨어 예외를 실행해야 합니다.
- 클러스터에 할당하는 인터페이스에 대한 것과 동일한 인터페이스 컨피그레이션을 포함해야 합니다(예: EtherChannel, 활성 인터페이스, 속도 및 이중 등). 동일한 인터페이스 ID에 대해 용량이 일치하고 동일한 Spanned EtherChannel에서 성공적인 인터넷 번들링이 가능한 한 새시에서 서로 다른 네트워크 모듈 유형을 사용할 수 있습니다. 모든 데이터 인터페이스는 새시 간 클러스터링에서 EtherChannel이어야 합니다.
- 동일한 NTP 서버를 사용해야 합니다. Firepower Threat Defense의 경우 Firepower Management Center도 동일한 NTP 서버를 사용해야 합니다. 시간을 수동으로 설정해서는 안 됩니다.
- ASA를 사용할 때 라이선스 기관에 등록해야 합니다. Firepower Threat Defense의 경우 모든 라이선싱이 Firepower Management Center에서 처리됩니다.

새시 간 클러스터링을 위한 스위치 사전 요구 사항

- FXOS 새시에서 클러스터링을 구성하기 전에 스위치 컨피그레이션을 완료하고 새시의 모든 EtherChannel을 스위치에 성공적으로 연결하십시오.
- 지원되는 스위치의 목록은 [Cisco FXOS 호환성](#)을 참고하십시오.

## 클러스터링 지침

### 모델

- Firepower 9300의 ASA - 인트라 새시, 새시 간 및 사이트 간 클러스터링에 지원됨.
- Firepower 4100 Series의 ASA - 새시 간 및 사이트 간 클러스터링에 지원됨.
- Firepower 9300의 Firepower Threat Defense - 인트라 새시 및 새시 간 클러스터링에 지원됨.
- Firepower 4100 Series의 Firepower Threat Defense - 새시 간 클러스터링에 지원됨.
- Radware DefensePro — ASA와의 인트라 새시 클러스터링(intra-chassis clustering)에 지원됨.

### 새시 간 클러스터링을 위한 스위치

- ASR 9006의 경우 기본이 아닌 MTU를 설정하려면 ASR 인터페이스 MTU를 디바이스 MTU보다 14바이트 높게 설정합니다. 그렇지 않으면, **mtu-ignore** 옵션을 사용하지 않는 경우 OSPF 인접 피어링 시도에 실패할 수 있습니다. 디바이스 MTU는 ASR IPv4 MTU와 일치해야 합니다.
- 클러스터 제어 링크 인터페이스용 스위치의 경우, 클러스터 유닛에 연결된 스위치 포트에서 Spanning Tree PortFast를 사용하도록 선택하여 새 유닛에 대한 참가 프로세스 속도를 높일 수 있습니다.
- 스위치에서 Spanned EtherChannel의 번들링 속도가 저하될 경우, 스위치의 개별 인터페이스에 대한 LACP 속도를 빠르게 설정할 수 있습니다.
- 스위치에서는 **source-dest-ip** 또는 **source-dest-ip-port** EtherChannel 로드 밸런싱 알고리즘 중 하나를 사용하는 것이 좋습니다(Cisco Nexus OS 및 Cisco IOS **port-channel load-balance** 명령 참조). 로드 밸런싱 알고리즘에서는 **vlan** 키워드를 사용하지 마십시오. 이렇게 할 경우 클러스터의 디바이스에 트래픽이 균일하지 않게 분산될 수 있습니다. 디바이스에서 로드 밸런싱 알고리즘의 기본값을 변경하지 마십시오.
- 스위치에서 EtherChannel의 로드 밸런싱 알고리즘을 변경할 경우, 스위치의 EtherChannel 인터페이스에서 트래픽 전달이 일시적으로 중단되며 Spanning Tree Protocol이 재시작됩니다. 트래픽에서 흐름을 다시 시작하기 전까지 지연이 발생하게 됩니다.
- Cisco Nexus 스위치의 경우 모든 클러스터용 EtherChannel 인터페이스에서 LACP Graceful Convergence 기능을 사용하지 않도록 설정해야 합니다.
- 클러스터 제어 링크 경로의 네트워크 요소에서는 L4 체크섬을 확인하지 않습니다. 클러스터 제어 링크를 통해 리디렉션된 트래픽에는 올바른 L4 체크섬이 없습니다. L4 체크섬을 확인하는 스위치의 경우 트래픽이 감소하는 결과를 초래할 수 있습니다.
- 포트 채널 번들링 다운타임은 구성된 keepalive 기간을 초과하면 안 됩니다.
- Supervisor 2T EtherChannel에서 기본 해시 분산 알고리즘은 적응형입니다. VSS 설계에서 비대칭 트래픽을 방지하려면 디바이스에 연결된 포트 채널의 해시 알고리즘을 다음과 같이 변경하여 수정합니다.

```
router(config)# port-channel/dhash-distributionfixed
```

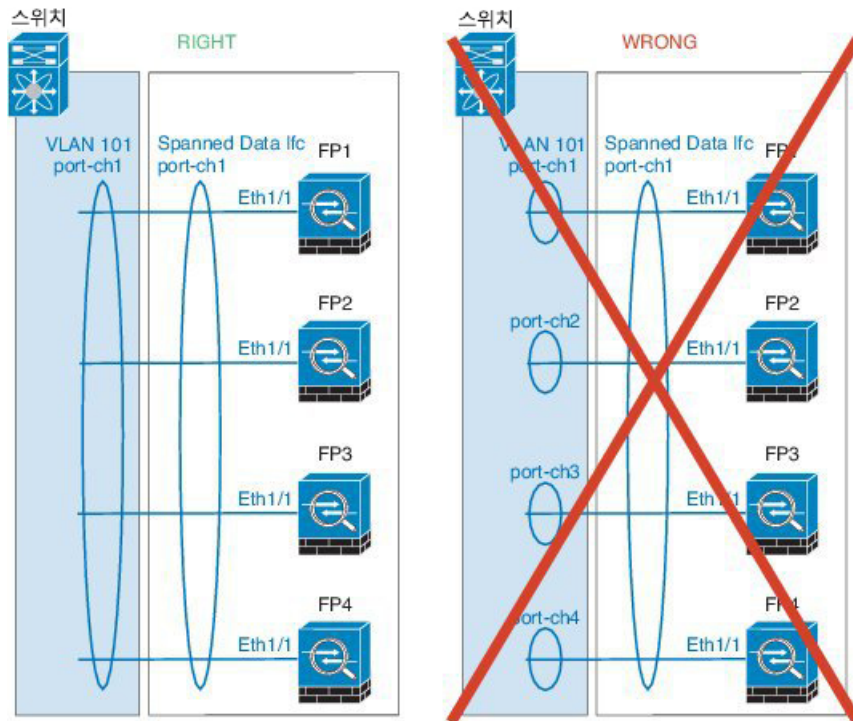
VSS 피어 링크의 적응형 알고리즘을 활용할 때가 있을 수 있으므로 알고리즘을 전역으로 변경하지 마십시오.

#### 새시 간 클러스터링을 위한 EtherChannel

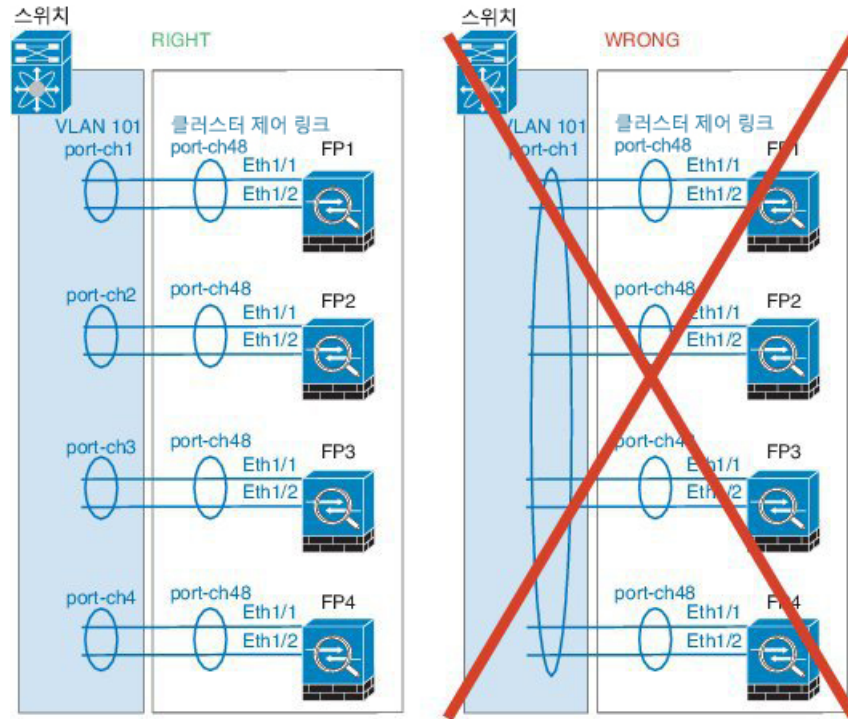
- 연결 스위치의 경우, EtherChannel 모드를 활성으로 설정합니다. On 모드는 FXOS 새시에서 지원되지 않으며 클러스터 제어 링크에서도 지원되지 않습니다.
- FXOS EtherChannel에서는 기본적으로 LACP 속도가 normal(표준)로 설정됩니다. 이 설정에서는 port-channel 멤버의 번들링에 30초가 넘게 걸릴 수 있으며, 이에 따라 클러스터 인터페이스 상태 확인이 실패하여 유닛이 클러스터에서 제거될 수 있습니다. LACP 속도를 fast(빠름)로 변경할 수 있습니다. 다음 예는 "기본" lacp 정책을 수정합니다.

```
firepower# scope org
firepower /org # scope lacppolicy default
firepower /org/lacppolicy# set lacp-rate fast
firepower /org* # commit-buffer
```

- 15.1(1)S2 이전 Catalyst 3750-X Cisco IOS 소프트웨어 버전에서는 클러스터 유닛에서 EtherChannel 과 스위치 스택 간 연결을 지원하지 않았습니다. 기본 스위치 설정으로 클러스터 유닛 EtherChannel 이 교차 스택에 연결되어 있는 상태에서 마스터 스위치의 전원이 꺼질 경우, 나머지 스위치에 연결된 EtherChannel은 가동되지 않습니다. 호환성을 개선하려면 **stack-mac persistent timer** 명령을 다시 로드 시간을 고려하여 충분히 큰 값으로 설정합니다(예: 8분 또는 무한인 경우 0). 또는 15.1(1)S2 같은 더 안정적인 스위치 소프트웨어 버전으로 업그레이드할 수 있습니다.
- Spanned EtherChannel 컨피그레이션과 디바이스-로컬 EtherChannel 컨피그레이션 — Spanned EtherChannel와 디바이스-로컬 EtherChannel에서 각각 알맞게 스위치를 구성해야 합니다.
  - Spanned EtherChannel — 클러스터의 모든 멤버 전체를 포괄하는 클러스터 유닛 *Spanned EtherChannels*의 경우, 인터페이스가 스위치의 단일 EtherChannel에 통합됩니다. 각 인터페이스가 스위치의 동일한 채널 그룹에 있는지 확인하십시오.



- 디바이스-로컬 EtherChannel — 클러스터 제어 링크에 대해 구성된 모든 EtherChannel을 비롯한 클러스터 유닛 디바이스-로컬 EtherChannel의 경우 스위치에서 별도의 EtherChannel을 구성해야 합니다. 여러 클러스터 유닛 EtherChannel을 스위치에서 하나의 EtherChannel에 통합하지 마십시오.



#### 추가 지침

- 이중화를 위해 EtherChannel을 VSS 또는 vPC에 연결하는 것이 좋습니다.
- 새시 내에서 일부 보안 모듈을 클러스터하여 독립형 모드에서 다른 보안 모듈을 실행할 수 없습니다. 클러스터에 모든 보안 모듈을 포함해야 합니다.

## 클러스터링 기본값

클러스터 제어 링크는 Port-channel 48을 사용합니다.

## ASA 클러스터링 구성

FXOS 새시 관리자(Supervisor)에서 손쉽게 클러스터를 구축할 수 있습니다. 모든 초기 컨피그레이션은 유닛마다 자동으로 생성됩니다. 새시 간 클러스터링의 경우 각 새시를 개별적으로 구성해야 합니다. 새시 하나에 클러스터를 구축한 다음 쉽게 구축하기 위해 첫 번째 새시의 부트스트랩 컨피그레이션을 다음 새시에 복사합니다.

#### 시작하기 전에

- 모듈을 설치하지 않은 경우에도 Firepower 9300 새시의 3개 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

- 멤버 인터페이스가 포함되지 않은 경우, **Interfaces**(인터페이스) 탭에서 port-channel 48 클러스터 유형 인터페이스에 **Operation State**(운영 상태)가 **failed**(실패)로 표시됩니다. 인트라 새시 클러스터링(intra-chassis clustering)의 경우 이 EtherChannel에는 멤버 인터페이스가 필요하지 않으므로 이 Operation State(운영 상태)를 무시할 수 있습니다.

## 절차

**단계 1** 클러스터를 구축하기 전에 데이터 유형 인터페이스 또는 EtherChannel(포트 채널)을 최소 1개 구성합니다. [포트 채널 생성, 146 페이지](#) 또는 [인터페이스 속성 편집, 145 페이지](#)를 참조하십시오. 모든 인터페이스는 클러스터에 기본적으로 할당되어 있습니다. 또한 데이터 인터페이스를 구축한 후에 클러스터에 추가할 수 있습니다.

새시 간 클러스터링의 경우, 모든 데이터 인터페이스는 멤버 인터페이스가 최소 1개 있는 EtherChannel 이어야 합니다. 각 새시에 EtherChannel을 추가합니다.

**단계 2** 관리 유형 인터페이스 또는 EtherChannel을 추가합니다. [포트 채널 생성, 146 페이지](#) 또는 [인터페이스 속성 편집, 145 페이지](#)를 참조하십시오.

**단계 3** Port-channel 48은 클러스터 제어 링크로 예약됩니다. 새시 간 클러스터링의 경우, 멤버 인터페이스 최소 1개를 port-channel 48에 추가합니다.

**단계 4** 보안 서비스 모드를 입력합니다.  
**Scope ssa**

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 새시 관리자(Supervisor)가 클러스터링 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 클러스터 이름이 아닙니다. 하드웨어를 아직 설치하지 않은 경우에도 보안 모듈 3개를 모두 지정해야 합니다.

**참고** 모듈을 설치하지 않은 경우에도 새시의 3개 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

**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 1자 ~ 38자로 된 ASCII 문자열이어야 합니다.

단계 8 클러스터 인터페이스 모드를 설정합니다.

```
set mode spanned-etherchannel
```

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set mode spanned-etherchannel
Firepower /ssa/logical-device/cluster-bootstrap* #
```

Spanned EtherChannel 모드는 유일하게 지원되는 모드입니다.

단계 9 관리 IP 주소 정보를 구성합니다.

이 정보는 보안 모듈 컨피그레이션의 관리 인터페이스를 구성하는 데 사용됩니다.

a) 로컬 IP 주소의 풀을 구성합니다. 이 중 하나는 인터페이스의 각 클러스터 유닛에 할당됩니다.

```
set ipv4 pool start_ip end_ip
```

```
set ipv6 pool start_ip end_ip
```

최소한 클러스터에 있는 유닛 수에 상응하는 개수의 주소를 포함해야 합니다. Firepower 9300에서는 모든 모듈 슬롯을 채우지 않은 경우에도 새시당 3개 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 새시 ID를 설정합니다.

**set chassis-id id**

클러스터의 각 새시에는 고유한 ID가 필요합니다.

예제:

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

단계 11 사이트 간 클러스터링의 경우 1~8의 사이트 ID를 설정합니다.

**set site-idnumber.**

예제:

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

단계 12 클러스터 제어 링크의 제어 트래픽에 대해 인증 키를 구성합니다.

**set key**

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set key
Key: diamonddogs
Firepower /ssa/logical-device/cluster-bootstrap* #
```

공유 비밀을 입력하라는 프롬프트가 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 13 클러스터 부트스트랩 모드 및 논리적 디바이스 모드를 종료합니다.

**exit**

**exit**

단계 14 사용 가능한 소프트웨어 버전을 확인한 다음 사용할 버전을 설정합니다.

a) 사용 가능한 버전을 표시합니다.

**show app**

예제:

```
/ssa # show app
```

```
Application:
  Name      Version  Description Author    Deploy Type  CSP Type  Is Default App
-----
  asa       9.1.4.152 N/A      cisco    Native      Application Yes
  asa       9.4.2    N/A      cisco    Native      Application No
```

```
asa          9.5.2.1    N/A          cisco       Native      Application No
```

b) 사용할 버전의 앱 모드를 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 15 컨피그레이션을 커밋합니다.

**commit-buffer**

FXOS 새시 관리자(Supervisor)는 기본 보안 모듈 소프트웨어 버전을 다운로드하고 클러스터 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 각 보안 모듈에 입력하여 클러스터를 구축합니다.

단계 16 클러스터에 다른 새시를 추가하려면 고유한 **chassis-id** 및 올바른 **site-id**를 구성해야 하는 경우를 제외하고 이 절차를 반복합니다. 아니면 두 새시 모두에 동일한 컨피그레이션을 사용합니다.

단계 17 클러스터링 컨피그레이션을 사용자 정의하려면 기본 유닛 보안 모듈에 연결합니다.

예

새시 1의 경우:

```
scope eth-uplink
scope fabric a
  enter port-channel 1
  set port-type data
  enable
  enter member-port Ethernet1/1
  exit
  enter member-port Ethernet1/2
  exit
  exit
  enter port-channel 2
  set port-type data
  enable
  enter member-port Ethernet1/3
  exit
  enter member-port Ethernet1/4
  exit
  exit
  enter port-channel 3
  set port-type data
  enable
  enter member-port Ethernet1/5
  exit
  enter member-port Ethernet1/6
  exit
  exit
```

```

enter port-channel 4
  set port-type mgmt
  enable
  enter member-port Ethernet2/1
  exit
  enter member-port Ethernet2/2
  exit
  exit
enter port-channel 48
  set port-type cluster
  enable
  enter member-port Ethernet2/3
  exit
  exit
  exit
exit
commit-buffer

scope ssa
  enter logical-device ASA1 asa "1,2,3" clustered
  enter cluster-bootstrap
    set chassis-id 1
    set ipv4 gateway 10.1.1.254
    set ipv4 pool 10.1.1.11 10.1.1.27
    set ipv6 gateway 2001:DB8::AA
    set ipv6 pool 2001:DB8::11 2001:DB8::27
    set key
    Key: f@arscape
    set mode spanned-etherchannel
    set service-type cluster1
    set virtual ipv4 10.1.1.1 mask 255.255.255.0
    set virtual ipv6 2001:DB8::1 prefix-length 64
  exit
  exit
scope app asa 9.5.2.1
  set-default
  exit
commit-buffer

```

새시 2의 경우:

```

scope eth-uplink
  scope fabric a
    create port-channel 1
      set port-type data
      enable
      create member-port Ethernet1/1
      exit
      create member-port Ethernet1/2
      exit
      exit
    create port-channel 2
      set port-type data
      enable
      create member-port Ethernet1/3
      exit
      create member-port Ethernet1/4
      exit
      exit
    create port-channel 3
      set port-type data
      enable
      create member-port Ethernet1/5
      exit
      create member-port Ethernet1/6
      exit
      exit
    create port-channel 4
      set port-type mgmt
      enable
      create member-port Ethernet2/1
      exit

```

```

        create member-port Ethernet2/2
        exit
    exit
    create port-channel 48
    set port-type cluster
    enable
    create member-port Ethernet2/3
    exit
    exit
    exit
    exit
    commit-buffer

scope ssa
    enter logical-device ASA1 asa "1,2,3" clustered
    enter cluster-bootstrap
    set chassis-id 2
    set ipv4 gateway 10.1.1.254
    set ipv4 pool 10.1.1.11 10.1.1.15
    set ipv6 gateway 2001:DB8::AA
    set ipv6 pool 2001:DB8::11 2001:DB8::19
    set key
    Key: f@rscape
    set mode spanned-etherchannel
    set service-type cluster1
    set virtual ipv4 10.1.1.1 mask 255.255.255.0
    set virtual ipv6 2001:DB8::1 prefix-length 64
    exit
    exit
scope app asa 9.5.2.1
    set-default
    exit
    commit-buffer

```

## Firepower Threat Defense 클러스터링 구성

FXOS 새시 관리자(Supervisor)에서 손쉽게 클러스터를 구축할 수 있습니다. 모든 초기 컨피그레이션은 유닛마다 자동으로 생성됩니다. 새시 간 클러스터링의 경우 각 새시를 개별적으로 구성해야 합니다. 새시 하나에 클러스터를 구축한 다음 쉽게 구축하기 위해 첫 번째 새시의 부트스트랩 컨피그레이션을 다음 새시에 복사합니다.

시작하기 전에

- 모듈을 설치하지 않은 경우에도 Firepower 9300 새시의 3개 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.
- 멤버 인터페이스가 포함되지 않은 경우, **Interfaces**(인터페이스) 탭에서 port-channel 48 클러스터 유형 인터페이스에 **Operation State**(운영 상태)가 **failed**(실패)로 표시됩니다. 인트라 새시 클러스터링(intra-chassis clustering)의 경우 이 EtherChannel에는 멤버 인터페이스가 필요하지 않으므로 이 Operation State(운영 상태)를 무시할 수 있습니다.

절차

- 
- 단계 1** 클러스터를 구축하기 전에 데이터 유형 인터페이스 또는 EtherChannel(포트 채널)을 최소 1개 구성합니다. [포트 채널 생성, 146 페이지](#) 또는 [인터페이스 속성 편집, 145 페이지](#)를 참조하십시오. 또한 데이터 인터페이스를 구축한 후에 클러스터에 추가할 수 있습니다.

새시 간 클러스터링의 경우, 모든 데이터 인터페이스는 멤버 인터페이스가 최소 1개 있는 EtherChannel 이어야 합니다. 각 새시에 EtherChannel을 추가합니다.

- 단계 2** (선택 사항) 클러스터를 구축하기 전에 Firepower 이벤트 처리 유형 인터페이스를 구성합니다. [인터페이스 속성 편집, 145 페이지](#)을 참조하십시오.  
이 인터페이스는 Firepower Threat Defense 디바이스의 보조 관리 인터페이스입니다. 이 인터페이스를 사용하려면 Firepower Threat Defense CLI에서 해당 IP 주소 및 기타 매개변수를 구성해야 합니다. 예를 들면 관리 트래픽을 이벤트(예: 웹 이벤트)에서 분리할 수 있습니다. Firepower Management Center 명령 참조에서 **configure network** 명령을 참조하십시오.
- 단계 3** 관리 유형 인터페이스 또는 EtherChannel을 추가합니다. [포트 채널 생성, 146 페이지](#) 또는 [인터페이스 속성 편집, 145 페이지](#)를 참조하십시오.
- 단계 4** Port-channel 48은 클러스터 제어 링크로 예약됩니다. 새시 간 클러스터링의 경우, 멤버 인터페이스 최소 1개를 port-channel 48에 추가합니다.
- 단계 5** 보안 서비스 모드를 입력합니다.

#### Scope ssa

예제:

```
Firepower # scope ssa
Firepower /ssa #
```

- 단계 6** 클러스터를 생성합니다.  
**enter logical-device *device\_name* ftd "1,2,3" clustered**

예제:

```
Firepower /ssa # enter logical-device FTD1 ftd "1,2,3" clustered
Firepower /ssa/logical-device* #
```

*device\_name*은 FXOS 새시 관리자(Supervisor)가 클러스터링 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 컨피그레이션에 사용되는 클러스터 이름이 아닙니다.

**참고** 모듈을 설치하지 않은 경우에도 새시의 3개 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

- 단계 7** 클러스터 부트스트랩 매개변수를 구성합니다.
- 클러스터 부트스트랩 객체를 생성합니다.  
**enter cluster-bootstrap**
  - 새시 ID를 설정합니다.  
**set chassis-id *id***  
클러스터의 각 새시에는 고유한 ID가 필요합니다.
  - 보안 모듈 컨피그레이션에서 클러스터 키를 설정합니다.  
**set key**  
공유 비밀을 입력하라는 프롬프트가 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.

- d) 클러스터 인터페이스 모드를 설정합니다.

**set mode spanned-etherchannel**

Spanned EtherChannel 모드는 유일하게 지원되는 모드입니다.

- e) 보안 모듈 컨피그레이션에서 클러스터 그룹 이름을 설정합니다.

**set service-type cluster\_name**

이름은 1자~38자로 된 ASCII 문자열이어야 합니다.

- f) 클러스터 부트스트랩 모드를 종료합니다.

**exit**

예제:

```
Firepower /ssa/logical-device* # enter cluster-bootstrap
Firepower /ssa/logical-device/cluster-bootstrap* # set chassis-id 1
Firepower /ssa/logical-device/cluster-bootstrap* # set key
Key: f@arscape
Firepower /ssa/logical-device/cluster-bootstrap* # set mode spanned-etherchannel
Firepower /ssa/logical-device/cluster-bootstrap* # set service-type cluster1
Firepower /ssa/logical-device/cluster-bootstrap* # exit
Firepower /ssa/logical-device/* #
```

## 단계 8 관리 부트스트랩 매개변수를 구성합니다.

- a) 관리 부트스트랩 객체를 생성합니다.

**entermgmt-bootstrapfd**

- b) 관리하는 Firepower Management Center의 IP 주소를 지정합니다.

**enterbootstrap-keyFIREPOWER\_MANAGER\_IP**

**setvalueIP\_address**

**exit**

- c) 논리적 디바이스가 작동할 모드(Routed(라우팅됨) 또는 Transparent(투명))를 지정합니다.

**enterbootstrap-keyFIREWALL\_MODE**

**setvaluefirewall\_mode**

**exit**

- d) 디바이스와 Firepower Management Center는 간에 공유할 키를 지정합니다.

**enterbootstrap-key-secretREGISTRATION\_KEY**

**setvalue**

*registration\_key*

**exit**

- e) 논리적 디바이스에 사용할 비밀번호를 지정합니다.

**enterbootstrap-key-secretPASSWORD**

**setvalue**

*password*

**exit**

- f) 논리적 디바이스의 정규화된 호스트 이름을 지정합니다.

**enterbootstrap-keyFQDN**

*setvaluefqdn*

**exit**

- g) 논리적 디바이스에서 사용할 쉼표로 구분된 DNS 서버 목록을 지정합니다.

**enterbootstrap-keyDNS\_SERVERS**

*setvaluedns\_servers*

**exit**

- h) 논리적 디바이스의 서버 도메인을 쉼표로 구분하여 지정합니다.

**enterbootstrap-keySEARCH\_DOMAINS**

*setvaluesearch\_domains*

**exit**

- i) 클러스터의 각 보안 모듈에 대해 관리 IP 주소를 구성합니다.

참고 Firepower 9300에서는 모듈을 설치하지 않은 경우에도 새시의 3개 모듈 슬롯 모두에 대해 IP 주소를 설정해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

다음과 같이 IPv4 관리 인터페이스 객체를 생성합니다.

- 1 관리 인터페이스 객체를 생성합니다.

**enteripv4slot\_idfirepower**

- 2 게이트웨이 주소를 설정합니다.

**setgatewaygateway\_address**

- 3 IP 주소 및 마스크를 설정합니다.

**setipip\_addressmasknetwork\_mask**

- 4 관리 IP 모드를 종료합니다.

**exit**

- 5 새시의 나머지 모듈에 대해 반복합니다.

다음과 같이 IPv6 관리 인터페이스 객체를 생성합니다.

- 1 관리 인터페이스 객체를 생성합니다.

**enteripv6slot\_idfirepower**

- 2 게이트웨이 주소를 설정합니다.

**setgatewaygateway\_address**

- 3 IP 주소 및 접두사를 설정합니다.

**set ipip\_addressprefix-lengthprefix**

## 4 관리 IP 모드를 종료합니다.

```
exit
```

## 5 새시의 나머지 모듈에 대해 반복합니다.

## j) 관리 부트스트랩 모드를 종료합니다.

```
exit
```

예제:

```
Firepower /ssa/logical-device* # enter mgmt-bootstrap ftd
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key FIREPOWER_MANAGER_IP
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 10.0.0.100
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key FIREWALL_MODE
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value routed
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key-secret REGISTRATION_KEY
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value: ziggy$ardust
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Value: $pidersfrommars
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key FQDN
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value example.cisco.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key DNS_SERVERS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value 192.168.1.1
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter bootstrap-key SEARCH_DOMAINS
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # set value example.com
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter ipv4 1 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.31 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter ipv4 2 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.32 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # enter ipv4 3 firepower
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.0.0.1
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.0.0.33 mask 255.255.255.0
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
Firepower /ssa/logical-device* #
```

## 단계 9 논리적 디바이스 모드를 종료합니다.

```
exit
```

## 단계 10 사용 가능한 소프트웨어 버전을 확인한 다음 사용할 버전을 설정합니다.

## a) 사용 가능한 버전을 표시합니다.

```
show app
```

예제:

```
/ssa # show app
Application:
```



Name	Version	Description	Author	Deploy Type	CSP Type	Is Default App
ftd	6.0.1.37	N/A	cisco	Native	Application	Yes
ftd	6.1.0.11	N/A	cisco	Native	Application	No
ftd	6.1.0.21	N/A	cisco	Native	Application	No

- b) 사용할 버전의 앱 모드를 입력합니다.  
**scope app ftdversion\_number**
- c) 이 버전을 기본값으로 설정합니다.  
**set-default**
- d) 이 버전의 최종 사용자 라이선스 계약에 동의합니다.  
**accept-license-agreement**
- e) 앱 모드를 종료합니다.  
**exit**

예제:

```
/ssa # scope app ftd 6.1.0.21
/ssa/app # set-default
/ssa/app* # accept-license-agreement
/ssa/app* # exit
/ssa* #
```

단계 11 컨피그레이션을 커밋합니다.

**commit-buffer**

FXOS 새시 관리자(Supervisor)는 기본 보안 모듈 소프트웨어 버전을 다운로드하고 클러스터 부트스트랩 컨피그레이션 및 관리 인터페이스 설정을 각 보안 모듈에 입력하여 클러스터를 구축합니다.

단계 12 클러스터에 다른 새시를 추가하려면 고유한 **chassis-id** 및 고유한 관리 IP 주소를 구성해야 하는 경우를 제외하고 이 절차를 반복합니다. 아니면 두 새시 모두에 동일한 컨피그레이션을 사용합니다.

단계 13 관리 IP 주소를 사용하여 각 보안 모듈을 Firepower Management Center에 추가한 다음 웹 인터페이스에서 클러스터로 그룹화합니다.

Firepower Management Center에 추가하기 전에 모든 클러스터 유닛이 FXOS에서 성공적으로 형성된 클러스터에 있어야 합니다.

예

```
scope eth-uplink
  scope fabric a
    enter port-channel 1
    set port-type data
    enable
    create member-port Ethernet1/1
    exit
    create member-port Ethernet1/2
    exit
    exit
  enter port-channel 2
  set port-type data
  enable
  create member-port Ethernet1/3
```

```

        exit
    create member-port Ethernet1/4
    exit
exit
enter port-channel 3
    set port-type firepower-eventing
    enable
    create member-port Ethernet1/5
    exit
    create member-port Ethernet1/6
    exit
    exit
enter port-channel 4
    set port-type mgmt
    enable
    create member-port Ethernet2/1
    exit
    enter member-port Ethernet2/2
    exit
    exit
enter port-channel 48
    set port-type cluster
    enable
    enter member-port Ethernet2/3
    exit
    exit
exit
exit
commit-buffer

scope ssa
    enter logical-device FTD1 ftd "1,2,3" clustered
    enter cluster-bootstrap
        set chassis-id 1
        set key cluster_key
        set mode spanned-etherchannel
        set service-type ftd-cluster
    exit
    enter mgmt-bootstrap ftd
        enter bootstrap-key FIREPOWER_MANAGER_IP
            set value 10.0.0.100
        exit
        enter bootstrap-key FIREWALL_MODE
            set value transparent
        exit
        enter bootstrap-key-secret REGISTRATION_KEY
            set value
            Value: alladinsane
        exit
        enter bootstrap-key-secret PASSWORD
            set value
            Value: widthofacircle
        exit
        enter bootstrap-key FQDN
            set value ftd.cisco.com
        exit
        enter bootstrap-key DNS_SERVERS
            set value 192.168.1.1
        exit
        enter bootstrap-key SEARCH_DOMAINS
            set value search.com
        exit
    enter ipv4 1 firepower
        set gateway 10.0.0.1
        set ip 10.0.0.31 mask 255.255.255.0
    exit
    enter ipv4 2 firepower
        set gateway 10.0.0.1
        set ip 10.0.0.32 mask 255.255.255.0
    exit
    enter ipv4 3 firepower
        set gateway 10.0.0.1
        set ip 10.0.0.33 mask 255.255.255.0

```

```

        exit
    exit
    exit
scope app ftd 6.0.0.837
    accept-license-agreement
    exit
commit-buffer

```

새시 2의 경우:

```

scope eth-uplink
    scope fabric a
        enter port-channel 1
            set port-type data
            enable
            create member-port Ethernet1/1
                exit
            create member-port Ethernet1/2
                exit
            exit
        enter port-channel 2
            set port-type data
            enable
            create member-port Ethernet1/3
                exit
            create member-port Ethernet1/4
                exit
            exit
        enter port-channel 3
            set port-type firepower-eventing
            enable
            create member-port Ethernet1/5
                exit
            create member-port Ethernet1/6
                exit
            exit
        enter port-channel 4
            set port-type mgmt
            enable
            create member-port Ethernet2/1
                exit
            enter member-port Ethernet2/2
                exit
            exit
        enter port-channel 48
            set port-type cluster
            enable
            enter member-port Ethernet2/3
                exit
            exit
        exit
    exit
commit-buffer

scope ssa
    enter logical-device FTD1 ftd "1,2,3" clustered
        enter cluster-bootstrap
            set chassis-id 2
            set key cluster_key
            set mode spanned-etherchannel
            set service-type ftd-cluster
            exit
        enter mgmt-bootstrap ftd
            enter bootstrap-key FIREPOWER_MANAGER_IP
                set value 10.0.0.100
                exit
            enter bootstrap-key FIREWALL_MODE
                set value transparent
                exit
            enter bootstrap-key-secret REGISTRATION_KEY
                set value
                Value: alladinsane

```

```

exit
enter bootstrap-key-secret PASSWORD
set value
Value: widthofacircle
exit
enter bootstrap-key FQDN
set value ftd.cisco.com
exit
enter bootstrap-key DNS_SERVERS
set value 192.168.1.1
exit
enter bootstrap-key SEARCH_DOMAINS
set value search.com
exit
enter ipv4 1 firepower
set gateway 10.0.0.1
set ip 10.0.0.31 mask 255.255.255.0
exit
enter ipv4 2 firepower
set gateway 10.0.0.1
set ip 10.0.0.32 mask 255.255.255.0
exit
enter ipv4 3 firepower
set gateway 10.0.0.1
set ip 10.0.0.33 mask 255.255.255.0
exit
exit
exit
scope app ftd 6.0.0.837
accept-license-agreement
exit
commit-buffer

```

## 클러스터링 기록

기능 이름	플랫폼 릴리스	기능 정보
Cisco ASA를 위한 새시 클러스터링	1.1.1	Firepower 9300 새시 내부에서 모든 ASA 보안 모듈을 클러스터링할 수 있습니다. 추가된 명령: <b>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</b>
ASA 모듈 6개를 위한 새시 간 클러스터링	1.1.3	현재 ASA를 위한 새시 간 클러스터링을 활성화할 수 있습니다. 최대 6개의 새시에 최대 6개의 모듈을 포함할 수 있습니다.
Firepower 9300의 Firepower Threat Defense에서 인트라 새시 클러스터링(intra-chassis clustering) 지원	1.1.4	Firepower 9300은 Firepower Threat Defense 애플리케이션이 있는 인트라 새시 클러스터링(intra-chassis clustering)을 지원합니다. 추가된 명령: <b>enter mgmt-bootstrap ftd, enter bootstrap-key FIREPOWER_MANAGER_IP, enter bootstrap-key FIREWALL_MODE, enter bootstrap-key-secret REGISTRATION_KEY, enter bootstrap-key-secret PASSWORD, enter bootstrap-key FQDN, enter bootstrap-key DNS_SERVERS, enter bootstrap-key SEARCH_DOMAINS, enter ipv4 firepower, enter ipv6 firepower, set value, set gateway, set ip, accept-license-agreement</b>

기능 이름	플랫폼 릴리스	기능 정보
FXOS 새시에서 ASA에 대한 사이트 간 클러스터링 개선	2.1.1	이제 ASA 클러스터를 구축할 때 각 FXOS 새시에 대한 사이트 ID를 구성할 수 있습니다. 전에는 ASA 애플리케이션 내에서 사이트 ID를 구성해야 했습니다. 이 기능 덕분에 초기 구축이 수월해졌습니다. 더 이상 ASA 컨피그레이션 내에서 사이트 ID를 설정할 수 없습니다. 또한 사이트 간 클러스터링과의 호환성을 최대한 활용하려면 안정성과 성능이 개선된 ASA 9.7(1) 및 FXOS 2.1.1로 업그레이드하는 것이 좋습니다.  다음 명령을 수정했습니다. <b>set site-id</b>
Firepower Threat Defense 모듈 6개를 위한 새시 간 클러스터링	2.1.1	이제 Firepower Threat Defense를 위한 새시 간 클러스터링을 활성화할 수 있습니다. 최대 6개의 새시에 최대 6개의 모듈을 포함할 수 있습니다.

## 서비스 체이닝 정보

Cisco FXOS 새시에서는 단일 블레이드에 있는 여러 서비스(예: 방화벽 및 서드파티 DDoS 애플리케이션)를 지원할 수 있습니다. 이 애플리케이션 및 서비스는 서비스 체인을 구성하기 위해 함께 연결될 수 있습니다. 현재 지원되는 서비스 체이닝 컨피그레이션에서 서드파티 Radware DefensePro 가상 플랫폼을 설치하여 ASA 방화벽 또는 Firepower Threat Defense 앞에서 실행할 수 있습니다. Radware DefensePro는 FXOS 새시에서 DDoS(Distributed Denial-of-Service) 탐지 및 완화 기능을 제공하는 KVM 기반 가상 플랫폼입니다. 서비스 체이닝이 FXOS 새시에서 활성화된 경우, 네트워크의 트래픽은 기본 ASA 또는 Firepower Threat Defense 방화벽에 도달하기 전에 먼저 DefensePro 가상 플랫폼을 통과해야 합니다.



## 참고

- Radware DefensePro 플랫폼은 다음 보안 어플라이언스에서 ASA와 함께 지원됩니다.
  - Firepower 9300
  - Firepower 4120
  - Firepower 4140
  - Firepower 4150
- Radware DefensePro 플랫폼은 다음 보안 어플라이언스에서 Firepower Threat Defense와 함께 지원됩니다.
  - Firepower 9300
  - Firepower 4110
  - Firepower 4120
  - Firepower 4140
  - Firepower 4150
- FXOS Chassis Manager를 사용하여 Radware DefensePro를 설치하는 것이 다음 시나리오에서 는 현재 지원되지 않습니다.
  - Firepower 4120 또는 4140 보안 어플라이언스에서 ASA와 함께 Radware DefensePro 실행
  - Firepower 9300 또는 4100 보안 어플라이언스에서 Firepower Threat Defense와 함께 Radware DefensePro 실행

위의 시나리오 중 하나에서 Radware DefensePro를 설치하는 경우 FXOS CLI를 사용하여 데코레이터를 구축해야 합니다.

- Firepower 4110 또는 4120 디바이스에서 Firepower Threat Defense 앞에 Radware DefensePro를 설치하는 경우 논리적 디바이스와 동시에 데코레이터를 구축해야 합니다. 논리적 디바이스가 이미 디바이스에 구성된 이후에는 데코레이터를 설치할 수 없습니다.
- Radware DefensePro 서비스 체인은 독립형 컨피그레이션 또는 ASA 방화벽 또는 Firepower Threat Defense가 있는 인트라 새시 클러스터링된 컨피그레이션에서 활성화할 수 있습니다.
- 서비스 체이닝은 새시 간 클러스터 컨피그레이션에서 지원되지 않습니다. 그러나 새시 간 클러스터 시나리오의 독립형 컨피그레이션에서는 Radware DefensePro 애플리케이션을 구축할 수 있습니다.
- DefensePro 애플리케이션은 최대 3개 보안 모듈에서 별도의 인스턴스로 실행될 수 있습니다.
- Radware DefensePro 가상 플랫폼은 *Radware vDP*(가상 DefensePro) 또는 간단하게 *vDP*라고도 합니다.
- Radware DefensePro 가상 플랫폼은 경우에 따라 링크 데코레이터라고도 합니다.
- FXOS 새시에 Radware DefensePro를 구축하기 전에 **etc/UTC** 표준 시간대로 NTP 서버를 사용하도록 FXOS 새시를 구성해야 합니다. FXOS 새시에서 날짜 및 시간을 설정하는 방법에 대한 자세한 내용은 [날짜 및 시간 설정, 103 페이지](#)를 참조하십시오.

## 독립형 논리적 디바이스에 Radware DefensePro 서비스 체인 구성

다음 절차는 Radware DefensePro 이미지를 설치하고 이 이미지를 독립형 ASA 또는 Firepower Threat Defense 논리적 디바이스의 앞에 있는 단일 서비스 체인에 구성하는 방법을 보여줍니다.

다음 절차는 Radware DefensePro 이미지를 설치하고 이 이미지를 독립형 ASA 또는 Firepower Threat Defense 논리적 디바이스의 앞에 있는 단일 서비스 체인에 구성하는 방법을 보여줍니다.

시작하기 전에

- 논리적 디바이스가 이미 구성되어 있는 보안 모듈을 논리적 디바이스에 사용하려는 경우, 기존의 논리적 디바이스를 먼저 삭제해야 합니다([논리적 디바이스 삭제, 188 페이지](#) 참고).
- Cisco.com에서 vDP 이미지를 다운로드([Cisco.com에서 이미지 다운로드, 48 페이지](#) 참고)한 다음 해당 이미지를 FXOS 새시에 다운로드합니다([FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드, 50 페이지](#) 참고).
- 포트 채널을 생성하고 인터페이스 유형을 {data|mgmt}로 설정합니다([포트 채널 생성, 146 페이지](#) 참고).

절차

- 단계 1** 독립형 컨피그레이션으로 ASA 논리적 디바이스 또는 Firepower Threat Defense 논리적 디바이스(독립형 ASA 논리적 디바이스 생성, 150 페이지 또는 독립형 위협 방어 논리적 디바이스 생성, 152 페이지 참조)를 생성합니다. Firepower 4110 또는 4120 보안 어플라이언스에 이미지를 설치하는 경우 Firepower Threat Defense 이미지와 함께 vDP 이미지를 설치해야 합니다.
- 단계 2** Radware vDP 이미지를 설치합니다.  
Firepower /ssa # **scopeslot\_id**  
Firepower /ssa/slot\_id # **createapp-instance vdp**
- 단계 3** 컨피그레이션을 커밋합니다.  
**commit-buffer**
- 단계 4** 보안 모듈의 vDP 설치 및 프로비저닝을 확인합니다.  
**show app-instance**
- 단계 5** ASA 또는 Firepower Threat Defense 논리적 디바이스를 입력합니다.  
**scope ssa**  
**scope logical-device { ld\_asa | ld\_ftd }**
- 단계 6** vDP용 외부 관리를 구성합니다.
  - a) 부트스트랩 객체를 생성합니다.  
**create mgmt-bootstrap vdp**



- b) 관리 IP 주소를 구성합니다.  
**createipv4slot\_iddefault**
- a) 게이트웨이 주소를 설정합니다.  
**setgatewaygateway\_address**
- b) IP 주소 및 마스크를 설정합니다.  
**setipip\_addressmasknetwork\_mask**
- c) 관리 IP 컨피그레이션 범위를 종료합니다.  
**exit**
- d) 관리 부트스트랩 컨피그레이션 범위를 종료합니다.  
**exit**

단계 7 외부 포트 링크를 생성합니다.  
**createexternal-port-linkmgmt\_vdpinterface\_id vdp**

단계 8 외부 포트의 범위를 표시합니다.  
**scopeexternal-port-linkdata\_1\_2**

단계 9 논리적 디바이스에 서드파티 앱을 추가합니다.  
**setdecorator vdp**  
**exit**  
**exit**

단계 10 서드파티 앱이 인터페이스에 설정되었는지 확인합니다.  
**show logical-device**

단계 11 컨피그레이션을 커밋합니다.  
**commit-buffer**

## 내장 새시 클러스터에 Radware DefensePro 서비스 체인 구성

다음 절차는 Radware DefensePro 이미지를 설치하고 이 이미지를 ASA 또는 Firepower Threat Defense 인프라 새시 클러스터 앞에 있는 서비스 체인에 구성하는 방법을 보여줍니다.

시작하기 전에

- 논리적 디바이스가 이미 구성되어 있는 보안 모듈을 논리적 디바이스에 사용하려는 경우, 기존의 논리적 디바이스를 먼저 삭제해야 합니다([논리적 디바이스 삭제](#), 188 페이지 참고).
- Cisco.com에서 vDP 이미지를 다운로드([Cisco.com에서 이미지 다운로드](#), 48 페이지 참고)한 다음 해당 이미지를 FXOS 새시에 다운로드합니다([FXOS 새시에 논리적 디바이스 소프트웨어 이미지 다운로드](#), 50 페이지 참고).

## 절차

- 단계 1 ASA 인프라 새시 클러스터(ASA 클러스터링 구성, 164 페이지 참조) 또는 Firepower Threat Defense 인프라 새시 클러스터(Firepower Threat Defense 클러스터링 구성, 170 페이지 참조)를 구성합니다.
- 단계 2 Radware DefensePro를 사용하여 외부(클라이언트 지향) 포트를 데코레이팅합니다.  
**enter external-port-linkname interface\_name { asa | ftd }**  
**set decoratorvdp**  
**set description ""**  
**exit**
- 단계 3 논리적 디바이스에 대한 외부 관리 포트를 할당합니다.  
**enter external-port-link { mgmt\_asa | mgmt\_ftd } interface\_name { asa | ftd }**  
**set decorator ""**  
**set description ""**  
**exit**
- 단계 4 DefensePro용 외부 관리 포트를 할당합니다.  
**enter external-port-linkmgmt\_vdp interface\_name { asa | ftd }**  
**set decorator ""**  
**set description ""**
- 단계 5 클러스터 포트 채널을 구성합니다.  
**enter external-port-link port-channel48 Port-channel48 { asa | ftd }**  
**set decorator ""**  
**set description ""**  
**exit**
- 단계 6 모든 DefensePro 인스턴스 3개에 대한 관리 부트스트랩을 구성합니다.  
**enter mgmt-bootstrap vdp**  
**enter ipv4 slot\_id default**  
**set gateway gateway\_address**  
**set ip\_address mask network\_mask**  
**exit**

## 예제:

```

enter mgmt-bootstrap vdp
  enter ipv4 1 default
    set gateway 172.16.0.1
    set ip 172.16.4.219 mask 255.255.0.0
  exit

  enter ipv4 2 default
    set gateway 172.16.0.1
    set ip 172.16.4.220 mask 255.255.0.0
  exit

```

```

enter ipv4 3 default
  set gateway 172.16.0.1
  set ip 172.16.4.221 mask 255.255.0.0
exit

```

단계 7 관리 부트스트랩 컨피그레이션 범위를 종료합니다.

**exit**

단계 8 마스터 블레이드에서 관리 IP를 설정하고 클러스터링을 활성화합니다.

**deviceclusteringmanagement-channelip**

**deviceclusteringmastersetmanagement-channelip**

**deviceclusteringstatesetenable**

단계 9 컨피그레이션을 커밋합니다.

**commit-buffer**

참고 이 절차를 완료한 후 DefensePro 인스턴스가 클러스터에 구성되었는지 확인해야 합니다.

단계 10 다음 방법 중 하나를 사용하여 어떤 DefensePro 인스턴스가 기본 또는 보조인지 확인합니다.

a) DefensePro 인스턴스 범위를 표시하고 DefensePro의 애플리케이션 특성만 보여줍니다.

**scope ssa**

**scope slot\_number**

**scope app-instance vdp**

**show app-attri**

b) 슬롯 범위를 표시하고 더 자세한 DefensePro 인스턴스 정보를 보여줍니다. 이 방식을 사용하면 슬롯에 있는 논리적 디바이스 및 vDP 애플리케이션 인스턴스의 정보를 모두 표시합니다.

**scope ssa**

**scope slot\_number**

**show app-instance expand detail**

DefensePro 애플리케이션이 온라인이지만 클러스터에 아직 구성되지 않은 경우 CLI는 다음을 표시합니다.

```

App Attribute:
App Attribute Key: cluster-role
Value: unknown

```

시스템이 “unknown” 값을 표시하면 DefensePro 애플리케이션을 시작하고 마스터 IP 주소를 구성하여 vDP 클러스터를 생성합니다.

DefensePro 애플리케이션이 온라인이며 클러스터에 구성되어 있는 경우 CLI는 다음을 표시합니다.

```

App Attribute:
App Attribute Key: cluster-role
Value: primary/secondary

```

예

```

scope ssa
  enter logical-device ld asa "1,2,3" clustered
    enter cluster-bootstrap
      set chassis-id 1
      set ipv4 gateway 172.16.0.1

```

```

        set ipv4 pool 172.16.4.216 172.16.4.218
        set ipv6 gateway 2010::2
        set ipv6 pool 2010::21 2010::26
        set key secret
        set mode spanned-etherchannel
        set name cisco
        set virtual ipv4 172.16.4.222 mask 255.255.0.0
        set virtual ipv6 2010::134 prefix-length 64
    exit
    enter external-port-link Ethernet1-2 Ethernet1/2 asa
        set decorator vdp
        set description ""
    exit
    enter external-port-link Ethernet1-3_asa Ethernet1/3 asa
        set decorator ""
        set description ""
    exit
    enter external-port-link mgmt_asa Ethernet1/1 asa
        set decorator ""
        set description ""
    exit
    enter external-port-link mgmt_vdp Ethernet1/1 vdp
        set decorator ""
        set description ""
    exit
    enter external-port-link port-channel48 Port-channel48 asa
        set decorator ""
        set description ""
    exit
    enter mgmt-bootstrap vdp
        enter ipv4 1 default
            set gateway 172.16.0.1
            set ip 172.16.4.219 mask 255.255.0.0
        exit

        enter ipv4 2 default
            set gateway 172.16.0.1
            set ip 172.16.4.220 mask 255.255.0.0
        exit

        enter ipv4 3 default
            set gateway 172.16.0.1
            set ip 172.16.4.221 mask 255.255.0.0
        exit
    exit
commit-buffer
scope ssa
    scope slot 1
    scope app-instance vdp
    show app-attri
    App Attribute:
    App Attribute Key: cluster-role
    Value: unknown

```

## UDP/TCP 포트 열기 및 vDP 웹 서비스 활성화

Radware APSolute Vision Manager 인터페이스는 다양한 UDP/TCP 포트를 사용하여 Radware vDP 애플리케이션과 통신합니다. vDP 애플리케이션이 APSolute Vision Manager와 통신하려면 이러한 포트에 액세스 가능하며 방화벽으로 인해 차단되지 않는지 확인해야 합니다. 열리는 특정 포트에 대한 자세한 내용은 APSolute Vision 사용 설명서의 다음 표를 참조하십시오.

- APSolute Vision Server-WBM 통신 및 운영 체제에 대한 포트
- Radware 디바이스를 사용하는 APSolute Vision Server의 통신 포트

Radware APSolute Vision에서 FXOS 새시에 구축된 가상 DefensePro 애플리케이션을 관리하려면 FXOS CLI를 사용하여 vDP 웹 서비스를 활성화해야 합니다.

### 절차

- 
- 단계 1** FXOS CLI에서 vDP 애플리케이션 인스턴스에 연결합니다.  
**connect moduleslot/console**  
**connect vdp**
- 단계 2** vDP 웹 서비스를 활성화합니다.  
**manage secure-web status set enable**
- 단계 3** vDP 애플리케이션 콘솔을 종료하고 FXOS 모듈 CLI로 돌아갑니다.  
**Ctrl ]**
- 

## 애플리케이션 콘솔 또는 데코레이터에 연결

다음 절차를 수행하여 애플리케이션 또는 데코레이터 콘솔에 연결합니다.



**참고** 콘솔 액세스 시 문제가 발생한 경우, 다른 SSH 클라이언트를 시도하거나 SSH 클라이언트를 새 버전으로 업그레이드하는 것이 좋습니다.

### 절차

- 
- 단계 1** 애플리케이션 또는 데코레이터 콘솔에 연결하려면 다음과 같이 합니다.
- FXOS CLI에서 보안 모듈/엔진에 연결합니다.  
**Firepower-chassis # connect module slot\_number console**  
**참고** 여러 보안 모듈을 지원하지 않는 디바이스의 보안 엔진에 연결하려면 *slot\_number*에 1을 사용합니다.  
 보안 모듈에 처음 연결할 때, FXOS 모듈 CLI에 액세스합니다.
  - 애플리케이션 또는 데코레이터에 연결하려면 디바이스에 적절한 명령을 입력합니다.  
**Firepower-module1>connect asa**  
**Firepower-module1>connect ftd**  
**Firepower-module1>connect vdp**  
 FXOS CLI의 관리자(Supervisor) 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.

FXOS 모듈 CLI에 대한 데코레이터 콘솔은 **Ctrl-]**를 입력하여 종료합니다.  
문제 해결을 위해 FXOS 모듈 CLI에 액세스할 수 있습니다.

- 단계 3 FXOS CLI의 관리자(Supervisor) 수준으로 돌아갑니다.
- 보안 모듈/엔진 콘솔을 종료하려면 ~를 입력합니다.  
텔넷 애플리케이션을 종료합니다.
  - 텔넷 애플리케이션을 종료하려면 다음을 입력합니다.  
**telnet>quit**

예

다음 예에서는 보안 모듈 1에 있는 ASA에 연결한 다음 FXOS CLI의 관리자(Supervisor) 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#
```

## 논리적 디바이스 삭제

절차

- 보안 서비스 모드를 입력합니다.  
**Firepower# scope ssa**
- 새시에 있는 논리적 디바이스에 대한 세부사항을 확인합니다.  
**Firepower /ssa # show logical-device**
- 삭제할 논리적 디바이스 각각에 대해 다음 명령을 입력합니다.  
**Firepower /ssa # delete logical-device device\_name**
- 논리적 디바이스에 설치되어 있는 애플리케이션에 대한 세부사항을 확인합니다.  
**Firepower /ssa # show app-instance**
- 삭제할 애플리케이션 각각에 대해 다음 명령을 입력합니다.
  - Firepower /ssa # scope slots slot\_number**
  - Firepower /ssa/slot # delete app-instance application\_name**

c) Firepower /ssa/slot # exit

단계 6 컨피그레이션을 커밋합니다.

**commit-buffer**

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa # show logical-device

Logical Device:
-----
Name          Description Slot ID      Mode          Operational State      Template Name
-----
FTD           1,2,3        Clustered    Ok                    ftd
Firepower /ssa # delete logical-device ftd
Firepower /ssa* # show app-instance
Application Name      Slot ID      Admin State      Operational State      Running Version Startup
Version Cluster Oper State
-----
ftd                   1 Disabled      Stopping          6.0.0.837
6.0.0.837            Not Applicable
ftd                   2 Disabled      Offline           6.0.0.837
6.0.0.837            Not Applicable
ftd                   3 Disabled      Not Available
6.0.0.837            Not Applicable
Firepower /ssa* # scope slot 1
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 2
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 3
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # commit-buffer
```

# 논리적 디바이스와 연결되지 않은 애플리케이션 인스턴스 삭제

논리적 디바이스를 삭제하면 논리적 디바이스의 애플리케이션 컨피그레이션도 삭제할 것인지 묻는 프롬프트가 표시됩니다. 애플리케이션 컨피그레이션을 삭제하지 않는 경우, 해당 애플리케이션 인스턴스를 삭제할 때까지 다른 애플리케이션을 사용하여 논리적 디바이스를 생성할 수 없습니다. 논리적 디바이스와 더 이상 연결되지 않은 애플리케이션 인스턴스를 보안 모듈/엔진에서 삭제하려면 다음 절차를 사용할 수 있습니다.

절차

단계 1 보안 서비스 모드를 입력합니다.

Firepower# **scope ssa**

단계 2 설치된 애플리케이션의 세부 사항을 봅니다.

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

단계 3 삭제할 애플리케이션 각각에 대해 다음 명령을 입력합니다.

- a) Firepower /ssa # **scope slotslot\_number**
- b) Firepower /ssa/slot # **delete app-instanceapplication\_name**
- c) Firepower /ssa/slot # **exit**

단계 4 컨피그레이션을 커밋합니다.

**commit-buffer**

시스템 컨피그레이션에 트랜잭션을 커밋합니다.

예

```
Firepower# scope ssa
Firepower /ssa* # show app-instance
Application Name      Slot ID      Admin State      Operational State      Running Version Startup
Version Cluster Oper State
-----
ftd                    1 Disabled      Stopping          6.0.0.837
6.0.0.837              Not Applicable
ftd                    2 Disabled      Offline           6.0.0.837
6.0.0.837              Not Applicable
ftd                    3 Disabled      Not Available
6.0.0.837              Not Applicable
Firepower /ssa* # scope slot 1
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 2
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # scope slot 3
Firepower /ssa/slot # delete app-instance ftd
Firepower /ssa/slot* # exit
Firepower /ssa* # commit-buffer
```





## 보안 모듈/엔진 관리

- [FXOS 보안 모듈/보안 엔진 정보, 191 페이지](#)
- [보안 모듈 해제/재위탁, 192 페이지](#)
- [보안 모듈/엔진 확인, 193 페이지](#)
- [보안 모듈/엔진 확인 재설정, 194 페이지](#)
- [보안 모듈/엔진 확인 다시 초기화, 195 페이지](#)
- [보안 모듈/엔진 전원 켜기/끄기, 196 페이지](#)

### FXOS 보안 모듈/보안 엔진 정보

보안 모듈/엔진에서 FXOS CLI를 사용하여 다음을 수행할 수 있습니다.

- **Decommission/Recommission(해제/재위탁)(보안 모듈만)** - 보안 모듈을 해제하면 보안 모듈이 유지 관리 모드로 들어갑니다. 또한 특정 장애 상태를 수정하려면 모듈을 해제한 후 재위탁할 수 있습니다. [보안 모듈 해제/재위탁, 192 페이지](#)을 참조하십시오.
- **Acknowledge(확인)** - 새로 설치된 보안 모듈을 온라인 상태로 전환합니다. [보안 모듈/엔진 확인, 193 페이지](#)을 참조하십시오.
- **Power Cycle(전력 사이클)** 보안 모듈/엔진을 재시작합니다. [보안 모듈/엔진 확인 재설정, 194 페이지](#)을 참조하십시오.
- **Reinitialize(다시 초기화)** - 보안 모듈/엔진 하드 디스크를 다시 포맷하여 모든 구축된 애플리케이션과 컨피그레이션을 보안 모듈/엔진에서 제거한 다음 시스템을 다시 시작합니다. 다시 초기화를 완료한 후, 보안 모듈/엔진에 대해 논리적 디바이스가 구성되어 있으면 Firepower eXtensible 운영 체제에서는 애플리케이션 소프트웨어를 다시 설치하고, 논리적 디바이스를 재구축하고, 애플리케이션을 자동으로 시작합니다. [보안 모듈/엔진 확인 다시 초기화, 195 페이지](#)을 참조하십시오.



**경고!** 보안 모듈/엔진의 모든 애플리케이션 데이터는 다시 초기화하는 동안 삭제됩니다. 보안 모듈/엔진을 다시 초기화하기 전에 모든 애플리케이션 데이터를 백업하십시오.

- 전원 끄기/켜기 - 보안 모듈/엔진의 전원 상태를 전환합니다. [보안 모듈/엔진 전원 켜기/끄기, 196 페이지](#)를 참조하십시오.

## 보안 모듈 해제/재위탁

보안 모듈을 해제하면, 보안 모듈 객체가 컨피그레이션에서 삭제되고 보안 모듈은 관리되지 않는 상태가 됩니다. 보안 모듈에서 실행되는 모든 논리적 디바이스 또는 소프트웨어는 비활성 상태가 됩니다.

보안 모듈의 사용을 일시적으로 중단하려는 경우 보안 모듈을 해제할 수 있습니다. 또한 보안 모듈을 다시 시작해도 오류 상태가 해결되지 않는 경우, 보안 모듈을 다시 초기화하지 않은 채 보안 모듈을 해제한 후 재위탁하여 오류 상태가 해결되는지 확인할 수 있습니다.

### 절차

**단계 1** 인터페이스 모드를 입력합니다.

```
scope eth-uplink
scope fabric a
```

**단계 2** 포트 채널을 생성합니다.

```
create port-channel id
활성화
```

**단계 3** 멤버 인터페이스를 할당합니다.

```
create member-port interface_id
```

예제:

```
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
9300 /eth-uplink/fabric/port-channel/member-port* # exit
```

**단계 4** (선택 사항) 인터페이스 유형을 설정합니다.

```
set port-type {data | mgmt | cluster}
```

예제:

```
9300 /eth-uplink/fabric/port-channel # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

**단계 5** (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.  
**set speed {10gbps | 1gbps}**

예제:

```
9300 /eth-uplink/fabric/port-channel* # set speed 1gbps
```

**단계 6** 컨피그레이션을 커밋합니다.  
**commit-buffer**

## 보안 모듈/엔진 확인

새로운 보안 모듈을 새시에 설치할 경우 먼저 보안 모듈을 확인해야만 사용할 수 있습니다.

보안 모듈의 상태가 "불일치" 또는 "토큰 불일치"로 나타나면, 슬롯에 새로 설치한 보안 모듈에 전에 설치했던 것과 일치하지 않는 데이터가 있음을 나타내는 것입니다. 보안 모듈에 기존의 데이터가 있고 이것을 새 슬롯에서 사용하려는 경우(다시 말하면, 보안 모듈을 실수로 잘못된 슬롯에 설치한 것이 아닌 경우), 여기에 논리적 디바이스를 구축하려면 먼저 보안 모듈을 다시 초기화해야 합니다.

절차

**단계 1** 인터페이스 모드를 입력합니다.  
**scope eth-uplink**  
**scope fabric a**

**단계 2** 포트 채널을 생성합니다.  
**create port-channel id**  
 활성화

**단계 3** 멤버 인터페이스를 할당합니다.  
**create member-port interface\_id**

예제:

```
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
9300 /eth-uplink/fabric/port-channel/member-port* # exit
```

```
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
9300 /eth-uplink/fabric/port-channel/member-port* # exit
```

단계 4 (선택 사항) 인터페이스 유형을 설정합니다.

```
set port-type {data | mgmt | cluster}
```

예제:

```
9300 /eth-uplink/fabric/port-channel # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

단계 5 (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.

```
set speed {10gbps | 1gbps}
```

예제:

```
9300 /eth-uplink/fabric/port-channel* # set speed 1gbps
```

단계 6 컨피그레이션을 커밋합니다.

```
commit-buffer
```

## 보안 모듈/엔진 확인 재설정

절차

단계 1 인터페이스 모드를 입력합니다.

```
scope eth-uplink
scope fabric a
```

단계 2 포트 채널을 생성합니다.

```
create port-channel id
```

활성화

단계 3 멤버 인터페이스를 할당합니다.

```
create member-port interface_id
```

예제:

```
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
```

```
9300 /eth-uplink/fabric/port-channel/member-port* # exit
```

단계 4 (선택 사항) 인터페이스 유형을 설정합니다.

```
set port-type {data | mgmt | cluster}
```

예제:

```
9300 /eth-uplink/fabric/port-channel # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

단계 5 (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.

```
set speed {10gbps | 1gbps}
```

예제:

```
9300 /eth-uplink/fabric/port-channel* # set speed 1gbps
```

단계 6 컨피그레이션을 커밋합니다.

```
commit-buffer
```

## 보안 모듈/엔진 확인 다시 초기화

보안 모듈/엔진을 다시 초기화하면 보안 모듈/엔진 하드 디스크가 포맷되고 설치된 모든 애플리케이션 인스턴스 및 컨피그레이션이 제거됩니다. 다시 초기화를 완료한 후, 보안 모듈/엔진에 대해 논리적 디바이스가 구성되어 있으면 Firepower eXtensible 운영 체제에서는 애플리케이션 소프트웨어를 다시 설치하고, 논리적 디바이스를 재구축하고, 애플리케이션을 자동으로 시작합니다.



**경고!** 보안 모듈/엔진의 모든 애플리케이션 데이터는 다시 초기화하는 동안 삭제됩니다. 보안 모듈/엔진을 다시 초기화하기 전에 모든 애플리케이션 데이터를 백업하십시오.

절차

단계 1 인터페이스 모드를 입력합니다.

```
scope eth-uplink  
scope fabric a
```

단계 2 포트 채널을 생성합니다.

```
create port-channel id  
활성화
```

단계 3 멤버 인터페이스를 할당합니다.

```
create member-port interface_id
```

예제:

```
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
9300 /eth-uplink/fabric/port-channel/member-port* # exit
```

단계 4 (선택 사항) 인터페이스 유형을 설정합니다.

**set port-type {data | mgmt | cluster}**

예제:

```
9300 /eth-uplink/fabric/port-channel # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

단계 5 (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.

**set speed {10gbps | 1gbps}**

예제:

```
9300 /eth-uplink/fabric/port-channel* # set speed 1gbps
```

단계 6 컨피그레이션을 커밋합니다.

**commit-buffer**

## 보안 모듈/엔진 전원 켜기/끄기

절차

단계 1 인터페이스 모드를 입력합니다.

**scope eth-uplink**  
**scope fabric a**

단계 2 포트 채널을 생성합니다.

**create port-channel id**  
활성화

단계 3 멤버 인터페이스를 할당합니다.

**create member-port interface\_id**

예제:

```
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/1
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/2
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/3
9300 /eth-uplink/fabric/port-channel/member-port* # exit
9300 /eth-uplink/fabric/port-channel* # create member-port Ethernet1/4
9300 /eth-uplink/fabric/port-channel/member-port* # exit
```

단계 4 (선택 사항) 인터페이스 유형을 설정합니다.

**set port-type {data | mgmt | cluster}**

예제:

```
9300 /eth-uplink/fabric/port-channel # set port-type mgmt
```

**data** 키워드는 기본 유형입니다. 이 포트 채널을 기본값 대신 클러스터 제어 링크로 사용하려는 경우가 아니라면 **cluster** 키워드를 선택하지 마십시오.

단계 5 (선택 사항) 포트 채널의 모든 멤버에 대해 인터페이스 속도를 설정합니다.

**set speed {10gbps | 1gbps}**

예제:

```
9300 /eth-uplink/fabric/port-channel* # set speed 1gbps
```

단계 6 컨피그레이션을 커밋합니다.

**commit-buffer**







## 컨피그레이션 가져오기/내보내기

- 컨피그레이션 가져오기/내보내기 정보, 199 페이지
- 컨피그레이션 파일 내보내기, 200 페이지
- 자동 컨피그레이션 내보내기 예약, 201 페이지
- 컨피그레이션 내보내기 미리 알림 설정, 201 페이지
- 컨피그레이션 파일 가져오기, 201 페이지

### 컨피그레이션 가져오기/내보내기 정보

컨피그레이션 내보내기 기능을 사용하여 FXOS 새시의 논리적 디바이스 및 플랫폼 컨피그레이션 설정을 포함하는 XML 파일을 원격 서버로 내보낼 수 있습니다. 나중에 해당 컨피그레이션 파일을 가져와서 컨피그레이션 설정을 FXOS 새시에 빠르게 적용하여, 알려진 정상적인 컨피그레이션으로 돌아가거나 시스템 장애로부터 복구할 수 있습니다.

#### 지침 및 제한 사항

- 컨피그레이션 파일의 내용을 수정하지 마십시오. 컨피그레이션 파일을 수정하면 해당 파일을 사용한 컨피그레이션 가져오기가 실패할 수 있습니다.
- 애플리케이션 관련 컨피그레이션 설정은 컨피그레이션 파일에 포함되지 않습니다. 애플리케이션 관련 설정 및 컨피그레이션을 관리하려면 애플리케이션에서 제공하는 컨피그레이션 백업 도구를 사용해야 합니다.
- FXOS 새시에서 컨피그레이션을 가져오면 FXOS 새시에 있는 모든 기존의 컨피그레이션(논리적 디바이스 포함)이 삭제되고 가져오기 파일에 포함된 컨피그레이션으로 완전히 교체됩니다.
- 컨피그레이션을 가져올 경우 원래 컨피그레이션을 내보낸 동일한 FXOS 새시로만 가져오는 것이 좋습니다.
- 컨피그레이션을 가져오는 FXOS 새시의 플랫폼 소프트웨어 버전은 내보낼 때와 동일한 버전이어야 합니다. 버전이 다르면 가져오기 작업의 성공이 보장되지 않습니다. FXOS 새시를 업그레이드 또는 다운그레이드할 때마다 백업 컨피그레이션을 내보내는 것이 좋습니다.

- 컨피그레이션을 가져오는 FXOS 새시에는 내보냈을 때와 동일한 슬롯에 동일한 네트워크 모듈이 설치되어 있어야 합니다.
- 컨피그레이션을 가져오는 FXOS 새시에는, 가져오는 내보내기 파일에 정의된 논리적 디바이스에 대해 올바른 소프트웨어 애플리케이션 이미지가 설치되어 있어야 합니다.
- 애플리케이션에 EULA(End-User License Agreement)가 있는 논리적 디바이스가 가져오는 컨피그레이션 파일에 포함되어 있으면, 컨피그레이션을 가져오기 전에 FXOS 새시에서 해당 애플리케이션의 EULA에 동의해야 합니다. 아니면 작업이 실패합니다.
- 기존 백업 파일을 덮어쓰지 않으려면 백업 작업 시 파일 이름을 변경하거나 기존 파일을 다른 위치에 복사하십시오.

## 컨피그레이션 파일 내보내기

FXOS 새시의 논리적 디바이스 및 플랫폼 컨피그레이션 설정을 포함하는 XML 파일을 원격 서버로 내보내려면 컨피그레이션 내보내기 기능을 사용합니다.

컨피그레이션 내보내기 기능 사용에 대한 중요한 정보는 [컨피그레이션 가져오기/내보내기 정보](#)를 참조하십시오.

### 절차

단계 1 예약된 내보내기 작업을 생성하려면:

```
scopeorg
scope cfg-export-policy export_task_name
set hostname hostname
```

참고 파일 이름을 포함한 전체 경로를 지정해야 합니다. 파일 이름을 지정하지 않으면 지정된 경로에 숨김 파일이 생성됩니다.

예제:

```
# scope org
# scope cfg-export-policy default
# set hostname 172.29.191.79
# set remote-file /data2/zhyin/export/export-config
# set protocol scp
# set user zhyin
# set password
Password:

# set schedule daily
# set adminstate enable
# commit-buffer
```

단계 2 내보내기 작업의 상태를 확인하려면:

```
scopesystem
scope export-config hostname
show fsm status
```

## 자동 컨피그레이션 내보내기 예약

FXOS 새시의 논리적 디바이스 및 플랫폼 컨피그레이션 설정을 포함하는 XML 파일을 원격 서버로 자동으로 내보내려면 예약된 내보내기 기능을 사용합니다. 내보내기를 매일, 매주 또는 2주마다 실행하도록 예약할 수 있습니다. 컨피그레이션 내보내기는 예약된 내보내기 기능이 활성화된 시기를 기반으로 예약에 따라 실행됩니다. 예를 들어 매주 수요일 오후 10시에 내보내기를 예약한 경우 시스템은 수요일마다 오후 10시에 새로운 내보내기를 트리거합니다.

컨피그레이션 내보내기 기능 사용에 대한 중요한 정보는 [컨피그레이션 가져오기/내보내기 정보](#)를 참조하십시오.

절차

## 컨피그레이션 내보내기 미리 알림 설정

특정 기간(일수)에 컨피그레이션 내보내기가 실행되지 않은 경우 시스템에서 오류를 생성하도록 하려면 Export Reminder(내보내기 미리 알림) 기능을 사용합니다.

절차

## 컨피그레이션 파일 가져오기

FXOS 새시에서 전에 내보낸 컨피그레이션 설정을 적용하려면 컨피그레이션 가져오기 기능을 사용할 수 있습니다. 이 기능을 사용하면 알려진 양호한 컨피그레이션으로 돌아가거나 시스템 장애로부터 복구할 수 있습니다. 컨피그레이션 가져오기 기능 사용에 대한 중요한 정보는 [컨피그레이션 가져오기/내보내기 정보](#)를 참조하십시오.

절차





# 14 장

## 패킷 캡처

- [패킷 캡처 정보, 203 페이지](#)
- [패킷 캡처 세션 생성 또는 수정, 204 페이지](#)
- [패킷 캡처에 대한 필터 구성, 206 페이지](#)
- [패킷 캡처 세션 시작 및 중지, 208 페이지](#)
- [패킷 캡처 파일 다운로드, 208 페이지](#)
- [패킷 캡처 세션 삭제, 209 페이지](#)

## 패킷 캡처 정보

패킷 캡처는 연결 및 컨피그레이션 문제를 디버깅하고 FXOS 새시를 통과하는 트래픽 흐름을 파악하기 위해 사용할 수 있는 매우 유용한 자산입니다. 패킷 캡처 도구를 사용하면 FXOS 새시의 특정 고객 대면 포트 또는 애플리케이션 포트를 통과하는 트래픽을 로깅할 수 있습니다.

여러 패킷 캡처 세션을 만들 수 있으며, 각 세션은 여러 포트의 트래픽을 캡처할 수 있습니다. 패킷 캡처 세션에 포함된 각 포트에 대해 별도의 패킷 캡처(PCAP) 파일이 생성됩니다.

### 백플레인 포트 매핑

FXOS 새시는 내부 백플레인 포트에 다음 매핑을 사용합니다.

보안 모듈	포트 매핑	설명
보안 모듈 1/검색 엔진	Ethernet1/9	Internal-Data0/0
보안 모듈 1/검색 엔진	Ethernet1/10	Internal-Data1/0
보안 모듈 2	Ethernet1/11	Internal-Data0/0
보안 모듈 2	Ethernet1/12	Internal-Data1/0

보안 모듈	포트 매핑	설명
보안 모듈 3	Ethernet1/13	Internal-Data0/0
보안 모듈 3	Ethernet1/14	Internal-Data1/0

### 지침 및 제한 사항

패킷 캡처 도구의 제한 사항은 다음과 같습니다.

- 최대 100Mbps까지만 캡처할 수 있습니다.
- 패킷 캡처 세션을 실행하기 위해 사용할 저장 공간이 충분하지 않을 경우에도 패킷 캡처 세션을 만들 수 있습니다. 패킷 캡처 세션을 시작하기 전에 저장 공간이 충분한지 확인해야 합니다.
- 여러 활성 패킷 캡처 세션은 지원되지 않습니다.
- 내부 스위치의 인그레스 단계에서만 캡처합니다.
- 내부 스위치에서 이해할 수 없는 패킷(Security Group Tag 및 Network Service Header 패킷)에는 필터가 효과적이지 않습니다.
- 추상화는 지원되지 않습니다(예: 포트 채널 및 서비스 체인).



**참고** 포트 채널의 트래픽 캡처는 지원되지 않지만, 패킷 캡처 세션에서 포트 채널을 구성하는 개별 멤버 포트를 포함할 수 있습니다. 그러면 패킷 캡처 도구는 각 해당 멤버 포트에 대해 별도의 패킷 캡처 파일을 만듭니다.

- 캡처 세션이 활성 상태인 동안에는 PCAP 파일을 복사하거나 내보낼 수 없습니다.
- 패킷 캡처 세션을 삭제하면 해당 세션과 연결된 모든 패킷 캡처 파일도 삭제됩니다.

## 패킷 캡처 세션 생성 또는 수정

### 절차

**단계 1** 패킷 캡처 모드로 들어갑니다.

```
fxos-chassis # scope packet-capture
```

**단계 2** 새 패킷 캡처 세션을 생성하려면:

```
fxos-chassis /packet-capture # create session session_name
```

기존의 패킷 캡처 세션을 수정하려면:

```
fxos-chassis /packet-capture # enter session session_name
```

단계 3 이 패킷 캡처 세션을 사용할 버퍼 크기를 지정합니다.

```
fxos-chassis /packet-capture/session # set session-memory-usage session_size_in_megabytes
```

1~2048MB 범위에서 버퍼 크기를 지정해야 합니다.

단계 4 이 패킷 캡처 세션에서 캡처할 패킷의 길이를 지정합니다.

```
fxos-chassis /packet-capture/session # set session-pcap-snaplength session_snap_length_in_bytes
```

지정된 스냅 길이는 64~9006바이트여야 합니다. 세션 스냅 길이를 구성하지 않으면 기본 캡처 길이는 1518바이트입니다.

단계 5 이 패킷 캡처 세션에 포함해야 할 포트를 지정합니다. 여러 포트에서 캡처할 수 있으며, 동일한 패킷 캡처 세션 중에 고객 대면 포트와 애플리케이션 포트 모두에서 캡처할 수 있습니다. 세션에 포함된 각 포트에 대해 별도의 패킷 캡처 파일이 생성됩니다.

참고 패킷 캡처 세션에서 포트를 제거하려면 아래에 나열된 목록에서 **create** 대신 **delete**를 사용합니다.

a) 고객 대면 포트를 추가하려면:

```
fxos-chassis /packet-capture/session* # create {phy-port | phy-aggr-port} port_name
```

phy-port의 경우 *port\_name* 구문은 **Ethernet**<slot\_id>/<port\_id> or **Port-Channel**<number>입니다.

Breakout 케이블의 경우(phy-aggr-port), *port\_name* 구문은

**Ethernet**<slot\_id>/<port\_id>/<breakout\_port\_id>입니다.

b) 애플리케이션 포트를 추가하려면:

```
fxos-chassis /packet-capture/session* # create app_port security_module_slot_id link_name interface_name app_name
```

c) 위의 단계를 필요한 만큼 반복하여 원하는 모든 포트를 추가합니다.

단계 6 캡처되는 트래픽을 필터링하려면:

패킷 캡처 세션에 포함된 인터페이스에 필터를 적용할 수 있습니다. 필터 생성에 대한 지침은 [패킷 캡처에 대한 필터 구성, 206 페이지](#) 섹션을 참조하십시오.

a) 필터를 적용할 인터페이스의 범위를 입력합니다.

```
fxos-chassis /packet-capture/session* # scope {phy-port | phy-aggr-port} port_name
```

```
scope phy-port Ethernet<slot_id>/<port_id>
```

or

```
scope phy-aggr-port Ethernet<slot_id>/<port_id>/<breakout_port_id>
```

or

```
scope <security_module_slot_id> <link_name> <interface_name> <app_name>
```

b) 원하는 필터를 적용합니다.

```
fxos-chassis /packet-capture/session/{phy-port|phy-aggr-port|app-port}* # set {source-filter} filtername
```

참고 포트에서 필터를 제거하려면 **set source-filter ""**를 사용합니다.

c) 추가 필터를 적용하려면 위의 단계를 필요한 만큼 반복합니다.

단계 7 패킷 캡처 세션을 지금 시작하려면:

```
fxos-chassis /packet-capture/session* # enable
```

새로 만든 패킷 캡처 세션은 기본적으로 비활성화됩니다. 세션을 명시적으로 활성화하면 변경이 커밋될 때 패킷 캡처 세션이 활성화됩니다. 다른 세션이 이미 활성 상태일 때 세션을 활성화하면 오류가 생성됩니다. 이 세션을 활성화하려면 우선 이미 활성화된 패킷 캡처 세션을 비활성화해야 합니다.

단계 8 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
fxos-chassis /packet-capture/session* # commit-buffer
```

패킷 캡처 세션을 활성화하면 시스템에서 패킷 캡처를 시작합니다. 세션에서 PCAP 파일을 다운로드하려면 먼저 캡처를 중지해야 합니다.

```
fxos-chassis# scope packet-capture
fxos-chassis packet-capture # create session asalinside
fxos-chassis packet-capture/session # set session-memory-usage 256
fxos-chassis packet-capture/session* # create phy-port Ethernet3/1
fxos-chassis packet-capture/session* # create phy-aggr-port Ethernet2/1/1
fxos-chassis packet-capture/session* # create app-port 1 link1 Ethernet 1/1 asa
fxos-chassis packet-capture/session* # create filter interfacelvlan100
fxos-chassis packet-capture/filter* # set ivlan 100
fxos-chassis packet-capture/filter* # set srcIP 6.6.6.6
fxos-chassis packet-capture/filter* # set srcPort 80
fxos-chassis packet-capture/filter* # set destIP 10.10.10.10
fxos-chassis packet-capture/filter* # set destPort 5050
fxos-chassis packet-capture/filter* # exit
fxos-chassis packet-capture/session* # scope phy-port Ethernet3/1
fxos-chassis packet-capture/session/phy-port* # set src-filter interfacelvlan100
fxos-chassis packet-capture/session/phy-port* # exit
fxos-chassis packet-capture/session* # scope app-port 1 link1 Ethernet1/1 asa
fxos-chassis packet-capture/session/app-port* # set src-filter interfacelvlan100
fxos-chassis packet-capture/session/app-port* # exit
fxos-chassis packet-capture/session* # enable
fxos-chassis packet-capture/session* # commit-buffer
fxos-chassis packet-capture/session #
```

## 패킷 캡처에 대한 필터 구성

패킷 캡처 세션에 포함된 트래픽을 제한할 필터를 만들 수 있습니다. 패킷 캡처 세션을 생성하는 동안 특정 필터를 사용해야 하는 인터페이스를 선택할 수 있습니다.



참고

현재 실행 중인 패킷 캡처 세션에 적용되는 필터를 수정하거나 삭제하는 경우, 해당 세션을 비활성화한 후 다시 활성화해야 변경 내용이 적용됩니다.

절차

단계 1 패킷 캡처 모드로 들어갑니다.

```
fxos-chassis # scope packet-capture
```

단계 2 새 패킷 캡처 필터를 생성하려면:

```
fxos-chassis /packet-capture # create filter filter_name
```

기존의 패킷 캡처 필터를 수정하려면:

```
fxos-chassis /packet-capture # enter filter filter_name
```

기존의 패킷 캡처 필터를 삭제하려면:

```
fxos-chassis /packet-capture # delete filter filter_name
```



단계 3 하나 이상의 필터 속성을 설정하여 필터 세부 사항을 지정합니다.

```
fxos-chassis /packet-capture/filter* # set <filterprop filterprop_value
```

참고 IPv4 또는 IPv6 주소를 사용하여 필터링할 수 있지만, 동일한 패킷 캡처 세션에서 두 주소를 모두 필터링할 수는 없습니다.

표 9: 지원되는 필터 속성

ivlan	Inner VLAN ID(포트로 들어가는 동안 패킷의 vlan)
ovlan	Outer VLAN ID(FXOS 새시에 의해 추가된 vlan)
srcip	소스 IP 주소(IPv4)
destip	목적지 IP 주소(IPv4)
srcipv6	소스 IP 주소(IPv6)
destipv6	목적지 IP 주소(IPv6)
srcport	소스 포트 번호
destport	목적지 포트 번호
protocol	IP 프로토콜[10진수 형식의 IANA 정의 Protocol 값]
ethertype	이더넷 프로토콜 유형[10진수 형식의 IANA 정의 이더넷 프로토콜 유형 값. 예: IPv4 = 2048, IPv6 = 34525, ARP = 2054, SGT = 35081]
srcmac	소스 MAC 주소
destmac	목적지 MAC 주소

```
fxos-chassis# scope packet-capture
fxos-chassis packet-capture # create filter interfacelvlan100
fxos-chassis packet-capture/filter* # set ivlan 100
fxos-chassis packet-capture/filter* # set srcip 6.6.6.6
fxos-chassis packet-capture/filter* # set srcport 80
fxos-chassis packet-capture/filter* # set destip 10.10.10.10
fxos-chassis packet-capture/filter* # set destport 5050
fxos-chassis packet-capture/filter* # commit-buffer
```

## 패킷 캡처 세션 시작 및 중지

### 절차

단계 1 패킷 캡처 모드로 들어갑니다.

```
fxos-chassis # scope packet-capture
```

단계 2 시작하거나 중지할 패킷 캡처 세션의 범위를 입력합니다.

```
fxos-chassis /packet-capture # enter session session_name
```

단계 3 패킷 캡처 세션을 시작하려면:

```
fxos-chassis /packet-capture/session* # enable [append | overwrite]
```

참고 다른 세션이 실행 중인 동안에는 패킷 캡처 세션을 시작할 수 없습니다.

패킷 캡처 세션이 실행 중인 동안에는 트래픽이 캡처될 때 개별 PCAP 파일의 크기가 증가합니다. 버퍼 크기 제한에 도달하면 시스템이 패킷 삭제를 시작하고 Drop Count(삭제 수) 필드가 증가합니다.

단계 4 패킷 캡처 세션을 중지하려면:

```
fxos-chassis /packet-capture/session* # disable
```

단계 5 시스템 컨피그레이션에 트랜잭션을 커밋합니다.

```
fxos-chassis /packet-capture/session* # commit-buffer
```

패킷 캡처 세션을 활성화한 경우, 세션에 포함된 인터페이스의 PCAP 파일이 트래픽 수집을 시작합니다. 세션 데이터를 덮어쓰도록 세션을 구성한 경우 기존 PCAP 데이터가 지워집니다. 아닌 경우 데이터가 기존 파일(있는 경우)에 추가됩니다.

```
fxos-chassis# scope packet-capture
fxos-chassis packet-capture # scope session asalinside
fxos-chassis packet-capture/session # enable append
fxos-chassis packet-capture/session* # commit-buffer
fxos-chassis packet-capture/session #
```

## 패킷 캡처 파일 다운로드

네트워크 패킷 분석기를 사용하여 분석할 수 있도록 세션에서 로컬 컴퓨터로 PCAP(Packet Capture) 파일을 다운로드할 수 있습니다.

PCAP 파일은 workspace://packet-capture 디렉터리에 저장되며 다음 명령 규칙을 사용합니다.

```
workspace://packet-capture/session-<id>/<session-name>-<interface-name>.pcap
```

### 절차

FXOS 새시에서 PCAP 파일을 복사하려면:

참고 패킷 캡처 세션에서 PCAP 파일을 다운로드하려면 먼저 해당 세션을 중지해야 합니다.

- a) 로컬 관리에 연결합니다.  
fxos-chassis # **connect localmgmt**
- b) PCAP 파일을 복사합니다.  
# **copy pcap\_file copy\_destination**

```
fxos-chassis# connect localmgmt
# copy workspace:/packet-capture/session-1/test-ethernet-1-1-0.pcap
scp://user@10.10.10.1:/workspace/
```

## 패킷 캡처 세션 삭제

현재 실행하고 있지 않은 개별 패킷 캡처 세션을 삭제하거나, 모든 비활성 패킷 캡처 세션을 삭제할 수 있습니다.

절차

- 
- 단계 1 패킷 캡처 모드로 들어갑니다.  
fxos-chassis # **scope packet-capture**
  - 단계 2 특정 패킷 캡처 세션을 삭제하려면:  
fxos-chassis /packet-capture # **delete session session\_name**
  - 단계 3 모든 비활성 패킷 캡처 세션을 삭제하려면:  
fxos-chassis /packet-capture # **delete-all-sessions**
  - 단계 4 시스템 컨피그레이션에 트랜잭션을 커밋합니다.  
fxos-chassis /packet-capture\* # **commit-buffer**
- 

```
fxos-chassis# scope packet-capture
fxos-chassis packet-capture # delete session asalinside
fxos-chassis packet-capture* # commit-buffer
fxos-chassis packet-capture #
```





## 색인

### A

- 객체 명령 [5](#)
- 계정 [33, 39, 40, 41, 44](#)
  - 로컬 인증 [33, 39, 40, 41, 44](#)
- 관리 IP 주소 [77](#)
  - 변경 [77](#)
- 관리 객체 [3](#)
- 구성 [118, 119, 120, 122, 123](#)
  - HTTPS [118, 119, 120, 122, 123](#)
- 기록, 비밀번호 [33](#)

### B

- 날짜 및 시간 [103](#)
  - 구성 [103](#)
- 논리적 디바이스 [53, 149, 150, 152, 161, 164, 170, 187, 188, 189](#)
  - 독립형 생성 [150, 152](#)
  - 삭제 [188](#)
  - 애플리케이션 인스턴스 삭제 [189](#)
  - 연결 [187](#)
  - 연결 종료 [187](#)
  - 이미지 버전 업데이트 [53](#)
  - 이해 [149](#)
    - 클러스터 생성 [161, 164, 170](#)
- 논리적 디바이스 연결 종료 [187](#)
- 논리적 디바이스에 연결 [187](#)
- 높은 수준의 작업 목록 [9](#)

### D

- date [104, 107](#)
  - 보기 [104](#)
  - 수동으로 설정 [107](#)
- DNS [140](#)

### E

- 명령 [6](#)
  - history [6](#)
- 명령 모드 [3](#)

### F

- 배너 [80, 81, 82](#)
  - pre-login [80, 81, 82](#)
- 보류 중인 명령 [7](#)
- 보안 모듈 [192, 193, 194, 195, 196](#)
  - 다시 초기화 [195](#)
  - 서비스 해제 [192](#)
  - 승인 [193](#)
  - 재설정 [194](#)
  - 전원 끄기 [196](#)
  - 전원 켜기 [196](#)
- 보안 모듈 다시 초기화 [195](#)
- 보안 모듈 재설정 [194](#)
- 보안 모듈 켜기/끄기 [196](#)
- 보안 모듈 해제 [192](#)
- 보안 모듈 확인 [193](#)
- 분할 케이블 [147](#)
  - 구성 [147](#)
- 분할 포트 [147](#)
- 비밀번호 [29, 33, 34, 37](#)
  - 강도 검사 [37](#)
  - 기록 수 [33](#)
  - 변경 간격 [34](#)
  - 지침 [29](#)
- 비밀번호 강도 적용 [37](#)
- 비밀번호 프로필 [33, 39, 40, 41, 44](#)
  - 변경 간격 [39](#)
  - 변경 간격 없음 [40](#)
  - 비밀번호 기록 수 [41](#)
  - 비밀번호 기록 지우기 [44](#)

비밀번호 프로필 (계속)  
정보 33

## G

사용자 7, 28, 29, 33, 34, 36, 37, 39, 40, 41, 43, 44, 116, 117

CLI 세션 제한 7

SNMP 116, 117

관리 28

기본 인증 34

로컬 인증 33, 39, 40, 41, 44

명명 지침 29

비밀번호 강도 검사 37

비밀번호 지침 29

비활성화 44

삭제 43

생성 41

역할 33

원격, 역할 정책 36

활성화 44

사용자 계정 33, 39, 40, 41, 44

비밀번호 프로필 33, 39, 40, 41, 44

새시 10

초기 컨피그레이션 10

소프트웨어 장애 89

복구 중 89

손상된 파일 시스템 93

복구 중 93

시스템 복구 89, 93

신뢰 지침 117, 122, 127

삭제 127

생성 122

정보 117

## H

원격 사용자에게 대한 역할 정책 36

위협 방어 152, 161, 170, 187, 188, 189

논리적 디바이스 삭제 188

독립형 위협 방어 논리적 디바이스 생성 152

애플리케이션 인스턴스 삭제 189

연결 187

연결 종료 187

클러스터 생성 161, 170

위협 방어 이미지 50

Firepower Security Appliance에 다운로드 50

이미지 47, 48, 49, 50

Cisco.com에서 다운로드 48

Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드 50

Firepower Security Appliance에 다운로드 48, 50

관리 47

무결성 확인 49

이미지 버전 53

업데이트 53

인증 34

기본 34

인터페이스 145

구성 145

속성 145

## I

작업 흐름 9

재부팅 83

정책 36

원격 사용자에게 대한 역할 36

## J

초기 컨피그레이션 10

## K

커뮤니티, SNMP 113

컨피그레이션 가져오기 199

컨피그레이션 가져오기/내보내기 199

제한 사항 199

지침 199

컨피그레이션 내보내기 199

클러스터 156, 161, 164, 170

생성 161, 164, 170

생성 시 기본값 164

정보 156

키 링 117, 118, 119, 120, 122, 123, 126

삭제 126

생성 118

신뢰 지침 122

인증서 가져오기 123

인증서 요청 119, 120

재생성 119

키 링 (계속)  
정보 117

## L

통신 서비스 113, 118, 119, 120, 122, 123  
HTTPS 118, 119, 120, 122, 123  
SNMP 113  
트랩 114, 115  
삭제 115  
생성 114

## M

패킷 캡처 203, 204, 206, 208, 209  
PCAP 파일 다운로드 208  
패킷 캡처 세션 삭제 209  
패킷 캡처 세션 생성 204  
패킷 캡처 세션 시작 208  
패킷 캡처 세션 중지 208  
필터 206  
패킷 캡처 세션 삭제 209  
패킷 캡처 세션 생성 204  
패킷 캡처 파일 다운로드 208  
펌웨어 54  
업그레이드 54  
펌웨어 업그레이드 54  
포트 채널 146  
구성 146  
표준 시간대 104, 106, 107  
설정 104, 106, 107  
프로필 33  
password 33  
플랫폼 번들 47, 48, 49, 50  
Cisco.com에서 다운로드 48  
Firepower Security Appliance에 다운로드 48  
무결성 확인 49  
업그레이드 50  
정보 47

## P

PCAP, 참조 패킷 캡처  
PCAP 파일 208  
다운로드 208  
PKI 117

pre-login 배너 80, 81, 82  
삭제 82  
생성 80  
수정 81

## R

RADIUS 133, 134, 135  
RADIUS 제공자 134, 135  
삭제 135  
생성 134  
rommon 54  
업그레이드 54  
RSA 117

## S

Smart Call Home 19  
HTTP 프록시 구성 19  
SNMP 110, 111, 112, 113, 114, 115, 116, 117  
Version 3 보안 기능 112  
권한 111  
보안 레벨 111  
사용자 116, 117  
삭제 117  
생성 116  
알림 111  
정보 110  
지원 110, 112  
커뮤니티 113  
트랩 114, 115  
삭제 115  
생성 114  
활성화 113  
SNMPv3 112  
보안 기능 112  
SSH 108  
구성 108  
syslog 138  
로컬 대상 구성 138  
로컬 소스 구성 138  
원격 대상 구성 138  
system 10  
초기 컨피그레이션 10

**T**

TACACS+ [136, 137](#)  
TACACS+ 제공자 [136, 137](#)  
    삭제 [137](#)  
    생성 [136](#)  
Telnet [109](#)  
    구성 [109](#)  
time [104, 107](#)  
    보기 [104](#)  
    수동으로 설정 [107](#)  
trap [111](#)  
    정보 [111](#)

**W**

라이선스 [20](#)  
    등록 [20](#)  
라이선스 등록 [20](#)  
로컬 인증 사용자 [33, 39, 40, 41, 44](#)  
    변경 간격 [39](#)  
    변경 간격 없음 [40](#)  
    비밀번호 기록 수 [41](#)  
    비밀번호 기록 지우기 [44](#)  
    비밀번호 프로필 [33](#)  
활성화 [113](#)  
SNMP [113](#)