



Cisco Firepower 9300 FXOS CLI 구성 가이드, 1.1(3)

초판: 2015년 12월 9일

최종 변경: 2016년 4월 27일

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 없음으로 인한 모든 파생적, 부수적, 직접, 간접, 특별, 징벌적 또는 기타 모든 손해(영업 이익 손실, 영업 중단, 영업 정보 손실, 또는 그 밖의 급전적 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. 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. (1721R)

© 2015 Cisco Systems, Inc. 모든 권리 보유.



목 차

1 장	Firepower Security Appliance 소개 1
	Firepower Security Appliance 정보 1

2 장	CLI 개요 3
	관리 객체 3
	명령 모드 3
	객체 명령 5
	명령 완성 6
	명령 기록 6
	보류 중인 명령 커밋, 삭제 및 보기 6
	CLI에 대한 인라인 도움말 7
	CLI 세션 제한 7

3 장	시작하기 9
	작업 흐름 9
	초기 구성 9
	액세스 - FXOS CLI 12

4 장	ASA의 라이선스 관리 15
	Smart Software Licensing 정보 15
	ASA의 Smart Software Licensing 16
	Smart Software Manager 및 어카운트 16
	오프라인 관리 16
	위성 서버 16

가상 어카운트별로 관리되는 라이선스 및 디바이스 17

평가판 라이선스 17

Smart Software Manager 통신 17

 디바이스 등록 및 토큰 17

 License Authority와의 정기적인 통신 18

 규정 위반 상태 18

 Smart Call Home 인프라 18

Smart Software Licensing 사전 요구 사항 18

스마트 소프트웨어 라이선싱을 위한 지침 19

Smart Software Licensing의 기본값 19

일반 Smart Software 라이선싱 구성 19

 (선택 사항) HTTP 프록시 구성 20

 License Authority에 Firepower Security Appliance 등록 21

Smart License Satellite Server 구성 Firepower 9300 새시 22

Smart Software Licensing 모니터링 23

Smart Software Licensing 기록 24

5 장 **사용자 관리 25**

 사용자 계정 25

 사용자 이름 지침 26

 비밀번호 지침 27

 원격 인증에 대한 지침 28

 사용자 역할 31

 로컬 인증 사용자에게 대한 비밀번호 프로파일 31

 기본 인증 서비스 선택 32

 세션 시간 초과 구성 34

 원격 사용자의 역할 정책 구성 35

 로컬로 인증된 사용자의 비밀번호 보안 수준 확인 활성화 36

 최대 로그인 시도 횟수 설정 36

 변경 간격에 대해 최대 비밀번호 변경 횟수 구성 37

 비밀번호에 대해 변경 안 함 간격 구성 38

비밀번호 기록 수 구성 39
 로컬 사용자 계정 생성 40
 로컬 사용자 계정 삭제 42
 로컬 사용자 계정 활성화 또는 비활성화 43
 로컬로 인증된 사용자의 비밀번호 기록 지우기 43

6 장

이미지 관리 45
 이미지 관리 정보 45
 Cisco.com에서 이미지 다운로드 46
 Firepower 9300 새시에 Firepower eXtensible 운영 체제 소프트웨어 이미지 다운로드 46
 이미지의 무결성 확인 47
 Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드 48
 논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 9300 새시 49
 논리적 디바이스를 위한 이미지 버전 업데이트 51
 펌웨어 업그레이드 53

7 장

플랫폼 설정 57
 관리 IP 주소 변경 57
 날짜 및 시간 설정 59
 표준 시간대 설정 59
 NTP를 사용하여 날짜 및 시간 설정 61
 NTP 서버 삭제 62
 날짜 및 시간 직접 설정 63
 SSH 구성 64
 텔넷 구성 65
 SNMP 구성 66
 SNMP 정보 66
 SNMP 알림 67
 SNMP 보안 수준 및 권한 67
 지원되는 SNMP 보안 모델과 수준 결합 67
 SNMPv3 보안 기능 68

SNMP 지원	68
SNMP 활성화 및 SNMP 속성 구성	69
SNMP 트랩 생성	70
SNMP 트랩 삭제	71
SNMPv3 사용자 생성	72
SNMPv3 사용자 삭제	74
HTTPS 구성	74
인증서, 키 링, 트러스트 포인트	75
키 링 생성	75
기본 키 링 재생성	76
키 링에 대한 인증서 요청 생성	77
기본 옵션으로 키 링에 대한 인증서 요청 생성	77
고급 옵션으로 키 링에 대한 인증서 요청 생성	78
트러스트 포인트 생성	80
키 링으로 인증서 가져오기	81
HTTPS 구성	83
HTTPS 포트 변경	84
키 링 삭제	85
트러스트 포인트 삭제	86
HTTPS 비활성화	86
AAA 구성	87
AAA 정보	87
LDAP 제공자 구성	88
LDAP 제공자의 속성 구성	88
LDAP 제공자 생성	89
LDAP 제공자 삭제	92
RADIUS 제공자 구성	93
RADIUS 제공자의 속성 구성	93
RADIUS 제공자 생성	94
RADIUS 제공자 삭제	95
TACACS+ 제공자 구성	96

TACACS+ 제공자의 속성 구성 96
 TACACS+ 제공자 생성 96
 TACACS+ 제공자 삭제 98
 Syslog 구성 98
 DNS 서버 구성 100

8 장

인터페이스 관리 103
 Firepower 인터페이스 정보 103
 새시 관리 인터페이스 103
 인터페이스 유형 103
 새시와 애플리케이션의 독립 인터페이스 상태 104
 Jumbo Frame Support 104
 Firepower 인터페이스에 대한 지침 및 제한 사항 104
 인터페이스 구성 105
 실제 인터페이스 구성 105
 EtherChannel(포트 채널) 추가 107
 분할 케이블 구성 109
 플로우 제어 정책 구성 110
 모니터링 인터페이스 112

9 장

논리적 디바이스 113
 논리적 디바이스 정보 113
 독립형 논리적 디바이스와 클러스터된 논리적 디바이스 113
 논리적 디바이스의 요구 사항 및 사전 요구 사항 114
 클러스터링의 요구 사항 및 사전 요구 사항 114
 논리적 디바이스 관련 지침 및 제한 사항 115
 일반 지침 및 제한 사항 116
 클러스터링 지침 및 제한 사항 116
 독립형 논리적 디바이스 추가 120
 독립형 ASA 추가 120
 고가용성 쌍 추가 125

- 클러스터 추가 126
 - 클러스터링 정보 Firepower 9300 새시 126
 - 기본 유닛 및 보조 유닛 역할 127
 - Cluster Control Link 127
 - 관리 네트워크 129
 - 관리 인터페이스 129
 - Spanned EtherChannels 129
 - 사이트 간 클러스터링 130
 - ASA 클러스터에 추가 131
 - ASA 클러스터 생성 131
 - 클러스터 멤버 더 추가 138
 - 논리적 디바이스 관리 138
 - 애플리케이션 콘솔에 연결 138
 - 논리적 디바이스 삭제 140
 - ASA를 투명 방화벽 모드로 변경 141
 - Firepower Threat Defense 논리적 디바이스의 인터페이스 변경 142
 - ASA 논리적 디바이스의 인터페이스 변경 144
 - 논리적 디바이스 모니터링 145
 - 사이트 간 클러스터링 예시 146
 - Spanned EtherChannel 투명 모드 노스-사우스 사이트 간의 예 146
 - Spanned EtherChannel 투명 모드 이스트-웨스트 사이트 간의 예 147
 - 논리적 디바이스의 기록 148

- 10 장 구성 가져오기/내보내기 149
 - 구성 가져오기/내보내기 정보 149
 - FXOS 구성 파일 내보내기 150
 - 자동 구성 내보내기 예약 152
 - 구성 내보내기 미리 알림 설정 153
 - 구성 파일 가져오기 154

- 11 장 패킷 캡처 157

패킷 캡처 157

- 백플레인 포트 매핑 157

패킷 캡처 관련 지침 및 제한 사항 158

패킷 캡처 세션 생성 또는 수정 158

패킷 캡처에 대한 필터 구성 160

패킷 캡처 세션 시작 및 중지 162

패킷 캡처 파일 다운로드 162



1 장

Firepower Security Appliance 소개

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

Firepower Security Appliance 정보

Cisco Firepower 9300 새시는 네트워크 및 콘텐츠 보안 솔루션을 위한 차세대 플랫폼입니다. Firepower 9300 새시는 Cisco ACI(Application Centric Infrastructure) 보안 솔루션에 포함되며 확장성, 제어 일관성 및 관리 간소화를 위해 구축된 민첩한 개방형 보안 플랫폼을 제공합니다.

Firepower 9300 새시에서 제공하는 기능은 다음과 같습니다.

- 모듈형 새시 기반 보안 시스템 — 고성능의 유연한 입/출력 구성 및 확장성을 제공합니다.
- Firepower Chassis Manager- 그래픽 사용자 인터페이스는 현재 새시 상태 및 새시 기능의 간소화된 구성을 간단하게 시각적으로 표시합니다.
- FXOS CLI — 기능 구성, 새시 상태 모니터링 및 고급 트러블슈팅 기능 액세스를 위해 명령어 기반 인터페이스를 제공합니다.
- FXOS REST API- 사용자가 새시를 프로그래밍 방식으로 구성 및 관리할 수 있습니다.



2 장

CLI 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, 6 페이지
- CLI에 대한 인라인 도움말, 7 페이지
- CLI 세션 제한, 7 페이지

관리 객체

FXOS(Firepower eXtensible 운영 체제)는 관리 객체 모델을 사용하며, 여기서 관리 객체는 관리 가능한 물리적 또는 논리적 엔티티를 추상화한 것입니다. 예를 들어, 새시, 보안 모듈, 네트워크 모듈, 포트 및 프로세서는 관리 객체로 표시된 물리적 엔티티이며 라이선스, 사용자 역할 및 플랫폼 정책은 관리 객체로 표시된 논리적 엔티티입니다.

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 **up** 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 top 명령	#
어댑터	EXEC 모드의 scope adapter 명령	/adapter #
케이블링	EXEC 모드의 scope cabling 명령	/cabling #
Chassis(새시)	EXEC 모드의 scope chassis 명령	/chassis #
이더넷 서버 도메인	EXEC 모드의 scope eth-server 명령. 이 명령과 모든 하위 명령은 현재 지원되지 않습니다.	/eth-server #
이더넷 업링크	EXEC 모드의 scope eth-uplink 명령	/eth-uplink #
Fabric Interconnect	EXEC 모드의 scope fabric-interconnect 명령	/fabric-interconnect #
펌웨어	EXEC 모드의 scope firmware 명령	/firmware #
호스트 이더넷 인터페이스	EXEC 모드의 scope host-eth-if 명령 참고 이 명령 및 모든 하위 명령은 이 레벨에서 지원되지 않습니다. 호스트 이더넷 인터페이스 명령은 /adapter # 모드에서 사용 가능합니다.	/host-eth-if #
라이선스	EXEC 모드의 scope license 명령	/license #

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

객체 명령

객체 관리에 사용 가능한 일반 명령 4개가 있습니다.

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

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

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

표 2: 객체가 없는 경우의 일반적인 동작

Command(명령)	행동
<code>create object</code>	객체가 생성되고 해당하는 경우 구성 모드가 시작됩니다.
<code>delete object</code>	오류 메시지가 생성됩니다.
<code>enter object</code>	객체가 생성되고 해당하는 경우 구성 모드가 시작됩니다.
<code>scope object</code>	오류 메시지가 생성됩니다.

표 3: 객체가 있는 경우의 일반적인 동작

Command(명령)	행동
<code>create object</code>	오류 메시지가 생성됩니다.
<code>delete object</code>	객체가 삭제됩니다.
<code>enter object</code>	해당하는 경우 객체의 구성 모드가 시작됩니다.
<code>scope object</code>	객체의 구성 모드가 시작됩니다.

명령 완성

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

명령 기록

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

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

보류 중인 명령 커밋, 삭제 및 보기

CLI에서 구성 명령어를 입력하면 **commit-buffer** 명령을 입력할 때까지 해당 명령이 적용되지 않습니다. 커밋될 때까지 구성 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 모든 보류 중인 명령의 유효성이 확인됩니다. 그러나 커밋 중에 대기된 명령 중 하나에서 장애가 발생하더라도 나머지 명령은 적용되며 장애가 발생한 명령은 오류 메시지에서 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 #
```

CLI에 대한 인라인 도움말

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

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

CLI 세션 제한

FXOS는 한 번에 활성화할 수 있는 CLI 세션의 수를 총 32개로 제한합니다. 이 값은 구성할 수 없습니다.



3 장

시작하기

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

작업 흐름

다음 절차에서는 Firepower 9300 새시 구성 시 완료해야 하는 기본 작업을 보여줍니다.

프로시저

- 단계 1 Firepower 9300 새시 하드웨어를 구성합니다([Cisco Firepower Security Appliance 하드웨어 설치 가이드 참조](#)).
- 단계 2 초기 구성을 완료합니다([초기 구성, 9 페이지 참조](#)).
- 단계 3 날짜 및 시간을 설정합니다([날짜 및 시간 설정, 59 페이지 참조](#)).
- 단계 4 DNS 서버를 구성합니다([DNS 서버 구성, 100 페이지 참조](#)).
- 단계 5 제품 라이선스를 등록합니다([ASA의 라이선스 관리, 15 페이지 참조](#)).
- 단계 6 사용자를 구성합니다([사용자 관리, 25 페이지 참조](#)).
- 단계 7 필요 시 소프트웨어 업데이트를 수행합니다([이미지 관리, 45 페이지 참조](#)).
- 단계 8 추가 플랫폼 설정을 구성합니다([플랫폼 설정, 57 페이지 참조](#)).
- 단계 9 인터페이스를 구성합니다([인터페이스 관리, 103 페이지 참조](#)).
- 단계 10 논리적 디바이스를 생성합니다([논리적 디바이스, 113 페이지 참조](#)).

초기 구성

Firepower Chassis Manager 또는 FXOS CLI를 사용하여 시스템을 구성하고 관리할 수 있으려면 먼저 콘솔 포트를 통해 액세스하는 FXOS CLI를 사용하여 초기 구성 작업 일부를 수행해야 합니다. FXOS

CLI를 사용하여 처음으로 Firepower 9300 새시에 액세스할 때 시스템을 구성하는 데 사용할 수 있는 설정 마법사가 나타납니다.

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

Firepower 9300 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. Firepower 9300 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)의 콘솔 포트 매개변수가 다음과 같은지 확인합니다.

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

프로시저

단계 1 콘솔 포트에 연결합니다.

단계 2 Firepower 9300 새시의 전원을 켭니다.

Firepower 9300 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.

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.

예

다음 예에서는 IPv4 관리 주소를 사용하여 구성을 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
Do you want to configure IP block for ssh access? (yes/no) [y]: y
  SSH IPv4 block netmask: 0.0.0.0
Do you want to configure IP block for https access? (yes/no) [y]: y
  HTTPS IP block address: 0.0.0.0
  HTTPS IPv4 block netmask: 0.0.0.0
Configure the DNS Server IP address (yes/no) [n]: y
  DNS IP address: 20.10.20.10
Configure the default domain name? (yes/no) [n]: y
  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
  SSH Access Configured=yes
    SSH IP Address=0.0.0.0
    SSH IP Netmask=0.0.0.0
  HTTPS Access Configured=yes
    HTTPS IP Address=0.0.0.0
    HTTPS IP Netmask=0.0.0.0
  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 구성을 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
Do you want to configure IP block for ssh access? (yes/no) [y]: y
  SSH IPv6 block netmask: 0.0.0.0
Do you want to configure IP block for https access? (yes/no) [y]: y
  HTTPS IP block address: 0.0.0.0
  HTTPS IPv6 block netmask: 0.0.0.0
Configure the DNS Server IPv6 address? (yes/no) [n]: y
  DNS IP address: 2001::101
Configure the DNS Server IP address (yes/no) [n]:
Configure the default domain name? (yes/no) [n]: y
  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
SSH Access Configured=yes
  SSH IP Address=0.0.0.0
  SSH IP Netmask=0.0.0.0
HTTPS Access Configured=yes
  HTTPS IP Address=0.0.0.0
  HTTPS IP Netmask=0.0.0.0
DNS Server=2001::101
Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes

```

액세스 - FXOS CLI

콘솔 포트에 전원이 연결된 터미널을 사용하여 FXOS CLI에 연결할 수 있습니다. 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 매개변수가 다음과 같은지 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 Firepower 9300 새시의 IP 주소 또는 호스트 이름을 알아야 합니다.

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



참고 SSH 로그인 은 대/소문자를 구분합니다.

SSH를 사용하는 Linux 터미널에서 다음 구문을 사용합니다.

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

```
ssh ucs-example\jsmith@192.0.20.11
ssh ucs-example\jsmith@2001::1
```
- **ssh -l ucs-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} -l ucs-auth-domain\username**

```
ssh 192.0.20.11 -l ucs-example\jsmith
```

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

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

```
ssh ucs-ldap23\jsmith@192.0.20.11
```

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

텔넷을 사용하는 Linux 터미널에서 다음 구문을 사용합니다.



참고 텔넷은 기본적으로 비활성화되어 있습니다. 텔넷 활성화에 대한 지침은 [텔넷 구성, 65 페이지](#)를 참고하십시오.

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

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

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

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

Putty 클라이언트에서 다음 구문을 사용합니다.

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

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



참고 기본 인증이 로컬로 설정되어 있고 콘솔 인증이 LDAP으로 설정된 경우, **ucs-local\admin**을 사용하여 Putty 클라이언트에서 패브릭 인터커넥트에 로그인할 수 있으며 이때 **admin**은 로컬 어카운트의 이름입니다.



4 장

ASA의 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.



참고 이 섹션은 Firepower 9300 새시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 환경 설정 가이드를 참조하십시오.

- [Smart Software Licensing 정보, 15 페이지](#)
- [Smart Software Licensing 사전 요구 사항, 18 페이지](#)
- [스마트 소프트웨어 라이선싱을 위한 지침, 19 페이지](#)
- [Smart Software Licensing의 기본값, 19 페이지](#)
- [일반 Smart Software 라이선싱 구성, 19 페이지](#)
- [Smart License Satellite Server 구성 Firepower 9300 새시, 22 페이지](#)
- [Smart Software Licensing 모니터링, 23 페이지](#)
- [Smart Software Licensing 기록, 24 페이지](#)

Smart Software Licensing 정보

이 섹션에서는 Smart Software Licensing이 적용되는 방법에 관해 설명합니다.



참고 이 섹션은 Firepower 9300 새시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 환경 설정 가이드를 참조하십시오.

ASA의 Smart Software Licensing

Firepower 9300 새시의 ASA 애플리케이션의 경우, Smart Software 라이선싱 구성은 Firepower 9300 새시 Supervisor(관리자)와 애플리케이션으로 나뉩니다.

- Firepower 9300 새시- 관리 프로그램에 모든 Smart Software 라이선싱 인프라를 구성하며 여기에는 License Authority와 통신하는 데 필요한 매개변수가 포함됩니다. Firepower 9300 새시 자체는 작동하기 위한 라이선스가 필요하지 않습니다.
- ASA 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.

오프라인 관리

디바이스가 인터넷에 액세스할 수 없어 License Authority에 등록할 수 없는 경우 오프라인 라이선싱을 구성할 수 있습니다.

위성 서버

디바이스가 보안상 인터넷에 액세스할 수 없는 경우 선택적으로 로컬 Smart Software Manager satellite 서버를 VM(가상 시스템)으로 설치할 수 있습니다. Smart Software Manager 기능의 하위 집합을 제공하는 위성을 통해 모든 로컬 디바이스에 대해 필수 라이선싱 서비스를 제공할 수 있습니다. 이 경우 라이선스 사용량을 동기화하기 위해 위성만 메인 License Authority에 주기적으로 연결하면 됩니다. 일정에 따라 동기화할 수도 있고 수동으로 동기화할 수도 있습니다.

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.

가상 어카운트별로 관리되는 라이선스 및 디바이스

라이선스 및 디바이스는 가상 어카운트별로 관리됩니다. 가상 계정의 디바이스에서 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.

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

평가판 라이선스

Firepower 9300 새시에서는 다음과 같은 두 가지 유형의 평가판 라이선스를 지원합니다.

- 새시 레벨 평가 모드 - Firepower 9300 새시는 Licensing Authority에 등록되기 전에 90일(총 사용량) 동안 평가 모드에서 작동합니다. 이 모드에서 ASA는 특정 엔타이틀먼트를 요청할 수 없으며 기본 엔타이틀먼트만 활성화됩니다. 이 기간이 종료되면 Firepower 9300 새시는 컴플라이언스 미준수 상태가 됩니다.
- 엔타이틀먼트 기반 평가 모드 - Firepower 9300 새시가 Licensing Authority에 등록되고 나면 ASA에 할당할 수 있는 시간 기반 평가판 라이선스를 받을 수 있습니다. ASA에서는 일반적인 방법으로 엔타이틀먼트를 요청합니다. 기간 기반 라이선스가 만료되면 해당 라이선스를 갱신하거나 영구 라이선스를 받아야 합니다.



참고 강력한 암호화(3DES/AES)용으로 평가판 라이선스를 받을 수는 없으며 영구 라이선스만 이 엔타이틀먼트를 지원합니다.

Smart Software Manager 통신

이 섹션에서는 디바이스가 Smart Software Manager와 통신하는 방법을 설명합니다.

디바이스 등록 및 토큰

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

구축 후 시작 시 또는 기존 새시에서 이 파라미터를 직접 구성한 이후에 새시는 Cisco License Authority에 등록됩니다. 토큰을 사용하여 새시가 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 Firepower 9300 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 구성을 변경할 수 없습니다. 이를 제외하면 작동에 영향을 미치지 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.

계정이 컴플라이언스 미준수 상태인지 또는 규정 위반 상태에 근접한지를 확인하려면 Firepower 9300 새시에서 현재 사용 중인 엔타이틀먼트와 Smart Account의 엔타이틀먼트를 비교해야 합니다.

컴플라이언스 미준수 상태에서는 특수 라이선스가 필요한 기능의 구성을 변경할 수는 없지만 작업은 달리 영향을 받지 않습니다. 예를 들어 표준 라이선스 한도를 초과하는 기존 컨텍스트를 계속 실행할 수 있으며 해당 구성을 수정할 수는 있지만 새 컨텍스트를 추가할 수는 없습니다.

Smart Call Home 인프라

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

Smart Software Licensing 사전 요구 사항

- 이 장은 Firepower 9300 새시의 ASA 논리적 디바이스에만 적용됩니다. Firepower Threat Defense 논리적 디바이스의 라이선싱에 대한 자세한 내용은 Firepower Management Center 환경 설정 가이드를 참조하십시오.
- 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 Commerce Workspace에서 라이선스를 1개 이상 구매합니다. 홈 페이지의 **Find Products and Solutions**(제품 및 솔루션 찾기) 검색 필드에서 사용 중인 플랫폼을 검색합니다. 일부 라이선스는 무료이지만 Smart Software 라이선싱 계정에 추가해야 합니다.
- 새시에서 Licensing Authority와 통신할 수 있도록 새시에서 인터넷 액세스 또는 HTTP 프록시 액세스를 보장합니다.
- 새시에서 Licensing Authority의 이름을 확인할 수 있도록 DNS 서버를 구성합니다.
- 새시의 시간을 설정합니다.
- ASA 라이선싱 엔타이틀먼트를 구성하기 전에 Firepower 9300 새시에 Smart Software 라이선싱 인프라를 구성합니다.

스마트 소프트웨어 라이선싱을 위한 지침

패일오버 및 클러스터링을 위한 **ASA** 지침

각 Firepower 9300 새시를 License Authority 또는 위성 서버에 등록해야 합니다. 보조 유닛에 대한 추가 비용은 없습니다.

Smart Software Licensing의 기본값

Firepower 9300 새시 기본 구성은 Smart Call Home 프로파일인 “SLProf”를 포함하며, 이는 Licensing Authority의 URL을 지정합니다.

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

일반 Smart Software 라이선싱 구성

Cisco License Authority와 통신하기 위해 HTTP 프록시를 선택적으로 구성할 수 있습니다. License Authority에 등록하려면 Smart Software 라이선스 계정에서 얻은 Firepower 9300 새시에 등록 토큰 ID를 입력해야 합니다.

프로시저

단계 1 (선택 사항) HTTP 프록시 구성, 20 페이지.

단계 2 License Authority에 Firepower Security Appliance 등록, 21 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.

프로시저

단계 1 HTTP 프록시를 활성화합니다.

scope monitoring scope callhome set http-proxy-server-enable on

예제:

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

단계 2 프록시 URL을 설정합니다.

set http-proxy-server-url url

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

예제:

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

단계 3 포트를 설정합니다.

set http-proxy-server-port port

예제:

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

단계 4 버퍼를 커밋합니다.

commit-buffer

License Authority에 Firepower Security Appliance 등록

Firepower 9300 새시를 등록할 때 License Authority에서는 Firepower 9300 새시와 License Authority의 통신을 위해 ID 인증서를 발급합니다. 또한 Firepower 9300 새시를 적절한 가상 계정에 지정합니다. 일반적으로 이 절차는 1회 수행됩니다. 그러나 이를테면 통신 문제 때문에 ID 인증서가 만료되면 나중에 Firepower 9300 새시를 다시 등록해야 할 수 있습니다.

프로시저

단계 1 Smart Software Manager 또는 Smart Software Manager Satellite에서 이 Firepower 9300 새시를 추가하려는 가상 계정에 대한 등록 토큰을 요청 및 복사합니다.

Smart Software Manager Satellite를 사용하여 등록 토큰을 요청하는 방법에 대한 자세한 내용은 Cisco Smart Software Manager Satellite 사용 설명서(http://www.cisco.com/web/software/286285517/138897/Smart_Software_Manager_satellite_4.1.0_User_Guide.pdf)를 참조하십시오.

단계 2 Firepower 9300 새시에 등록 토큰을 입력합니다.

scope license

register idtoken *id-token*

예제:

```
scope license
  register idtoken ZGFmNWM5NjgtYmNjYS00ZWl3LW
  WE3NGIttMWJkOGExZjIxNGQ0LTE0NjI2NDYx%0AMDIZNT
  V8N3R0dXM1Z0NjWkdpR214eFZhMldBOS9CVnNEYnVKM1
  g3R3dvemRD%0AY29NQTO%3D%0A
```

단계 3 이후에 디바이스의 등록을 취소하려면 다음을 입력합니다.

scope license

deregister

Firepower 9300 새시를 등록 취소하면 계정에서 디바이스가 제거됩니다. 디바이스의 모든 라이선스 엔타이틀먼트 및 인증서가 제거됩니다. 새 Firepower 9300 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 license

scope licdebug

renew

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

Smart License Satellite Server 구성 Firepower 9300 새시

다음 절차는 Smart License Manager Satellite를 사용하도록 Firepower 9300 새시를 구성하는 방법을 보여줍니다.

시작하기 전에

- [Smart Software Licensing 사전 요구 사항, 18 페이지](#)에 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.
- 인증서 체인이 아직 없는 경우 요청하려면 다음 절차를 수행합니다.
 - 키 링을 생성합니다([키 링 생성, 75 페이지](#)).
 - 해당 키 링에 대해 인증서 요청을 생성합니다([기본 옵션으로 키 링에 대한 인증서 요청 생성, 77 페이지](#)).
 - 이 인증서 요청을 Trust anchor 또는 인증 기관으로 전송하여 키 링용 인증서 체인을 받습니다.

자세한 내용은 [인증서, 키 링, 트러스트 포인트, 75 페이지](#)를 참고하십시오.

프로시저

단계 1 Callhome 대상으로 Satellite 서버를 설정합니다.

scope monitoring

scope call-home

scope profile SLProfile

scope destination SLDest

set address https://ip_address /Transportgateway/services/DeviceRequestHandler

단계 2 새 Trust Point를 생성합니다.

a) 보안 모드를 입력합니다.

scope security

b) Trust Point를 생성하고 이름을 지정합니다.

create trustpoint *trustpoint_name*

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

set certchain *certchain*

certchain 변수에는 이 절차의 인증서 생성 사전 요구 사항 수행 시에 받은 인증서 체인 정보를 사용합니다.

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

- d) 구성을 커밋합니다.

commit-buffer

예제:

```
firepower-chassis# scope security
firepower-chassis /security # create trustpoint tPoint10
firepower-chassis /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBASt
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCCyU
> ZgAMiVycsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKOND1
> GMBkPayVlQjbG4MD2dx2+H8EH3LmtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzC190306Mg51zq1zXcz75+VFj2I6rH9asckClD3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtv1WvfhevskV0j6
> jtcEMyZ+f7+3yh42lido3n04MIgeBgNVHSMGgZYwgZOAFLLnjtcEMyZ+f7+3yh42
> lido3n04oXikdjBOMQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAcT
> ClNhbncRhIENsYXJhMRswGQYDVQQKEwJ0dW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0Vuz2luZWVyaW5nMQ8wDQYDVQQDEw0ZXN0Q0GCAQAwdAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGqQxc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8cop1EBmOcyuhf5C6vasrennlDDkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIZJwK5+N3x/kfa2EHU6idlavt
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
firepower-chassis /security/trustpoint* # commit-buffer
firepower-chassis /security/trustpoint #
```

- 단계 3 License Authority에 Firepower 9300 새시를 등록합니다(License Authority에 Firepower Security Appliance 등록, 21 페이지 참조). Smart License Manager Satellite에서 등록 토큰을 요청하고 복사해야 합니다.

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
Firepower 9300 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 라이선싱 구성은 Firepower 9300 새시 Supervisor(관리자)와 보안 모듈로 나뉩니다.</p> <p>추가된 명령: deregister, register idtoken, renew, scope callhome, scope destination, scope licdebug, scope license, scope monitoring, scope profile, set address, set http-proxy-server-enable on, set http-proxy-server-url, set http-proxy-server-port, show license all, show license status, show license techsupport</p>



5 장

사용자 관리

- 사용자 계정, 25 페이지
- 사용자 이름 지침, 26 페이지
- 비밀번호 지침, 27 페이지
- 원격 인증에 대한 지침, 28 페이지
- 사용자 역할, 31 페이지
- 로컬 인증 사용자에게 대한 비밀번호 프로파일, 31 페이지
- 기본 인증 서비스 선택, 32 페이지
- 세션 시간 초과 구성, 34 페이지
- 원격 사용자의 역할 정책 구성, 35 페이지
- 로컬로 인증된 사용자의 비밀번호 보안 수준 확인 활성화, 36 페이지
- 최대 로그인 시도 횟수 설정, 36 페이지
- 변경 간격에 대해 최대 비밀번호 변경 횟수 구성, 37 페이지
- 비밀번호에 대해 변경 안 함 간격 구성, 38 페이지
- 비밀번호 기록 수 구성, 39 페이지
- 로컬 사용자 계정 생성, 40 페이지
- 로컬 사용자 계정 삭제, 42 페이지
- 로컬 사용자 계정 활성화 또는 비활성화, 43 페이지
- 로컬로 인증된 사용자의 비밀번호 기록 지우기, 43 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 어카운트

관리자 계정은 기본 사용자 계정이며 수정하거나 삭제할 수 없습니다. 이 어카운트는 시스템 관리자 또는 Superuser 어카운트이며 전체 권한을 가집니다. 관리자 어카운트에 할당된 기본 비밀번호가 없습니다. 초기 시스템 설정을 하는 동안 비밀번호를 선택해야 합니다.

관리자 어카운트는 항상 활성 상태이며 만료되지 않습니다. 관리자 어카운트는 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 구성 세부사항은 데이터베이스에 의해 삭제되지 않습니다. 비활성화된 로컬 사용자 계정을 다시 활성화하는 경우, 어카운트는 사용자 이름 및 비밀번호를 포함한 기존 구성으로 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, 28 페이지](#)
- [LDAP 제공자 구성, 88 페이지](#)
- [RADIUS 제공자 구성, 93 페이지](#)
- [TACACS+ 제공자 구성, 96 페이지](#)

사용자 계정 만료

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

기본적으로, 사용자 계정은 만료되지 않습니다.

만료일이 지정된 사용자 계정을 구성한 후에는 이 어카운트가 만료되지 않도록 재구성할 수 없습니다. 단, 최신 만료일이 있는 어카운트를 구성할 수 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 1~32자로 구성하며 다음을 포함할 수 있습니다.
 - 알파벳 문자
 - 숫자

- _(밑줄)
- -(대시)
- .(점)
- 로그인 ID는 고유해야 합니다.
- 로그인 ID는 알파벳 문자로 시작해야 합니다. 숫자 또는 밑줄과 같은 특수 문자로 시작할 수 없습니다.
- 로그인 ID는 대/소문자를 구분합니다.
- 모두 숫자인 로그인 ID를 생성할 수 없습니다.
- 사용자 계정을 생성한 후, 로그인 ID를 변경할 수 없습니다. 사용자 계정을 삭제하고 새로 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 길이 검사를 활성화하면 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.
- 다음 중 3개 이상을 포함해야 합니다.
 - 대문자 알파벳 문자
 - 소문자 알파벳 문자
 - 영숫자 외 문자(특수 문자)
 - 숫자
- 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 이름을 반대로 한 이름과 동일하지 않아야 합니다.
- 비밀번호 디ictionary 검사를 통과해야 합니다. 예를 들어, 비밀번호는 표준 사전 단어에 기반을 둘 수 없습니다.
- 다음 기호는 포함할 수 없습니다. 예: \$(달러 기호), ?(물음표) 및 =(등호)

- 로컬 사용자 및 관리자 계정 비밀번호는 비어 있지 않아야 합니다.

원격 인증에 대한 지침

지원되는 원격 인증 서비스 중 하나가 시스템에 구성될 경우, Firepower 9300 새시에서 시스템과 통신할 수 있도록 그 서비스에 대한 제공자를 생성해야 합니다. 다음 지침은 사용자 인증에 영향을 미칩니다.

원격 인증 서비스의 사용자 계정

사용자 계정은 Firepower 9300 새시의 로컬에 두거나 원격 인증 서버에 둘 수 있습니다.

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

원격 인증 서비스의 사용자 역할

원격 인증 서버에 사용자 계정을 생성할 경우 그 계정은 Firepower 9300 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 TACACS+ 구성에서는 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하는 원격 인증 제공자 각각에서 Firepower 9300 새시에 대한 사용자 속성을 구성해야 합니다. 이 사용자 특성은 각 사용자에게 지정된 역할 및 로케일을 저장합니다.

사용자가 로그인하면 FXOS에서 다음을 수행합니다.

1. 원격 인증 서비스를 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	맞춤형 속성	스키마 확장	속성 ID 요구 사항
LDAP	선택 사항	<p>다음 중 하나를 선택하여 수행할 수 있습니다.</p> <ul style="list-style-type: none"> LDAP 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 속성을 구성합니다. LDAP 스키마를 확장하고 CiscoAVPair와 같은 고유한 이름으로 맞춤형 속성을 생성합니다. 	<p>Cisco LDAP 구현에서는 유니코드 형식의 속성이 필요합니다.</p> <p>CiscoAVPair 맞춤형 속성을 생성하려는 경우 속성 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"> RADIUS 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 속성을 사용합니다. RADIUS 스키마를 확장하고 cisco-avpair와 같은 고유한 이름으로 맞춤형 속성을 생성합니다. 	<p>Cisco RADIUS 구현의 벤더 ID는 009, 속성의 벤더 ID는 001입니다.</p> <p>다음 구문의 예에서는 cisco-avpair 속성을 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	맞춤형 속성	스키마 확장	속성 ID 요구 사항
TACACS+	필수	스키마를 확장하고 cisco-av-pair 라는 이름으로 맞춤형 속성을 생성해야 합니다.	<p>cisco-av-pair 이름은 TACACS+ 제공자에 대한 속성 ID를 제공하는 문자열입니다.</p> <p>다음 구문의 예에서는 cisco-av-pair 속성을 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>cisco-av-pair 속성 구문에 별표(*)를 사용하면 로케일에 선택 사항 플래그를 지정합니다. 그러면 동일한 권한 부여 프로필을 사용하는 다른 Cisco 디바이스의 인증이 실패하지 않습니다. 여러 값을 구분하는 구분 기호로 공백을 사용합니다.</p>

LDAP 사용자 속성에 대한 샘플 OID

다음은 맞춤형 CiscoAVPair 속성에 대한 샘플 OID입니다.

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



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

사용자 역할

시스템에는 다음과 같은 사용자 역할이 포함됩니다.

관리자

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

읽기 전용

시스템 구성에 대한 읽기 전용 액세스로, 시스템 상태를 수정할 권한이 없습니다.

로컬 인증 사용자에 대한 비밀번호 프로파일

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

비밀번호 기록 수

비밀번호 기록 수를 사용하면 로컬로 인증된 사용자가 동일한 비밀번호를 계속해서 재사용하는 것을 방지할 수 있습니다. 이 속성을 구성할 때, Firepower 새시는 로컬로 인증된 사용자가 이전에 사용한 비밀번호를 최대 15개까지 저장합니다. 비밀번호는 최근 항목부터 시간순으로 저장되며, 기록 수 임계값에 도달할 경우 가장 오래된 비밀번호만 재사용될 수 있습니다.

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

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

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

비밀번호 변경 간격

비밀번호 변경 간격을 사용하면 로컬로 인증된 사용자가 지정된 시간 이내에 변경할 수 있는 비밀번호 변경 횟수를 제한할 수 있습니다. 다음 표는 비밀번호 변경 간격의 구성 옵션 2개를 설명합니다.

간격 구성	설명	예
비밀번호 변경 허용 안 됨	이 옵션을 사용하면 비밀번호 변경 이후 지정된 시간 동안 로컬로 인증된 사용자 비밀번호의 변경이 허용되지 않습니다. 변경 안 함 간격을 1~745시간으로 지정할 수 있습니다. 기본적으로, 변경 안 함 간격은 24시간입니다.	예를 들어, 로컬로 인증된 사용자가 비밀번호를 변경한 후 48시간 이내에 비밀번호가 변경되는 것을 방지하려면 다음을 설정합니다. <ul style="list-style-type: none"> • 해당 간격 동안 변경을 비활성화로 설정 • 변경 안 함 간격을 48시간으로 설정
변경 간격 내에 비밀번호 변경 허용됨	이 옵션은 로컬로 인증된 사용자가 미리 정의한 간격 동안 비밀번호를 변경할 수 있는 최대 횟수를 지정합니다. 변경 간격을 1~745시간으로 지정하고 비밀번호 변경 최대 횟수를 0~10으로 지정할 수 있습니다. 기본적으로, 로컬로 인증된 사용자는 48시간 동안 비밀번호 변경이 최대 2회 허용됩니다.	예를 들어, 로컬로 인증된 사용자가 비밀번호를 변경한 후 24시간 이내에 비밀번호를 최대 한 번 변경하도록 허용하려면 다음을 설정합니다. <ul style="list-style-type: none"> • 해당 간격 동안 변경을 활성화로 설정 • 변경 횟수를 1로 설정 • 변경 간격을 24로 설정

기본 인증 서비스 선택

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 기본 권한 부여 보안 모드를 입력합니다.

```
Firepower-chassis /security # scope default-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
```

60 ~ 172800의 정수로 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
```

60 ~ 172800의 정수로 지정합니다. 기본값은 7200초입니다.

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 구성에 트랜잭션을 커밋합니다.

```
commit-buffer
```

예

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 #
```

세션 시간 초과 구성

FXOS CLI를 사용하여 Firepower 9300 새시에서 사용자 세션을 종료할 때까지 사용자가 아무런 작업을 수행하지 않는 상태로 경과할 수 있는 시간을 지정할 수 있습니다. 콘솔 세션과 HTTPS/SSH/텔넷 세션에 대해 서로 다른 설정을 구성할 수 있습니다.

최대 3600초(60분)의 시간 초과 값을 설정할 수 있습니다. 기본값은 600초입니다. 이 설정을 비활성화하려면 세션 시간 초과 값을 0으로 설정합니다.

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 기본 권한 부여 보안 모드를 입력합니다.

```
Firepower-chassis /security # scope default-auth
```

단계 3 HTTPS, SSH 및 텔넷 세션에 대한 유희 시간 제한을 설정합니다.

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

단계 4 (선택 사항) 콘솔 세션에 대한 유희 시간 제한을 설정합니다.

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

단계 5 (선택 사항) 세션 및 절대 세션 시간 초과 설정을 봅니다.

```
Firepower-chassis /security/default-auth # show detail
```

예제:

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

원격 사용자의 역할 정책 구성

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

원격 사용자의 역할 정책을 다음 방법으로 구성할 수 있습니다.

assign-default-role

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

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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, 27 페이지 참조](#)).

프로시저

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}
```

예

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 Firepower 9300 새시를 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

- 기본 최대 로그인 시도 실패 횟수는 3입니다. 최대 로그인 시도 횟수를 초과한 후 사용자가 시스템에서 잠기는 기본 시간은 60분(3600초)입니다.

프로시저

단계 1 FXOS CLI에서 보안 모드로 들어갑니다.

scope system

scope security

단계 2 최대 로그인 시도 실패 횟수를 설정합니다.

set max-login-attempts

max_login

단계 3 최대 로그인 시도 횟수에 도달한 후 사용자가 시스템에서 잠긴 상태로 유지되는 시간(초)을 지정합니다.

set user-account-unlock-time

unlock_time

단계 4 구성을 커밋합니다.

commit-buffer

변경 간격에 대해 최대 비밀번호 변경 횟수 구성

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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

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

비밀번호에 대해 변경 안 함 간격 구성

프로시저

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
```

이 값은 1~745시간으로 선택할 수 있습니다.

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

단계 5 시스템 구성에 트랜잭션을 커밋합니다.


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

예

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

프로시저

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
```

이 값은 0~15로 선택할 수 있습니다.

기본적으로 **History Count**(기록 수) 필드가 0으로 설정되어 있어 기록 수가 비활성화되고 사용자가 언제든지 이전에 사용한 비밀번호를 재사용할 수 있습니다.

단계 4 시스템 구성에 트랜잭션을 커밋합니다.

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

예

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

프로시저

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, 26 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, 27 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 달 이름의 처음 세 글자입니다.

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

참고 만료일이 지정된 사용자 계정을 구성한 후에는 이 어카운트가 만료되지 않도록 재구성할 수 없습니다. 단, 최신 만료일이 있는 어카운트를 구성할 수 있습니다.

단계 8 (선택 사항) 사용자의 이메일 주소를 지정합니다.

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

단계 9 (선택 사항) 사용자 전화 번호를 지정합니다.

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

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

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

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

```
Firepower-chassis /security/local-user # create role role-name
```

여기서 *role-name*은 사용자 계정에 할당하고자 하는 권한을 나타내는 역할입니다([사용자 역할, 31 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 role role-name
```

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

단계 13 트랜잭션을 커밋합니다.

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

예

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/S1f30klCWjnV31gdXMzO0WU15iPw851kdQqap+NFuNmHcb4K
iaQB8X/PDdmtlxQQcawclj+k8f4VcOelBx1sGk5luq51s1ob1VOIEwcKEL/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/S1f30klCWjnV31gdXMzO0WU15iPw8
>51kdQqap+NFuNmHcb4KiaQB8X/PDdmtlxQQcawclj+k8f4VcOelBx1sGk5luq51s1ob1VO
>IEwcKEL/h51rdbN1I8y3SS9I/gGiBZ9ARlop9LDpDm8HPh2LOgyH7Ei1MI8=
> ---- END SSH2 PUBLIC KEY ----
> ENDOFBUF
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

로컬 사용자 계정 삭제

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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.

프로시저

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 활성 상태로 설정됩니다. 수정할 수 없습니다.

예

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
```

로컬로 인증된 사용자의 비밀번호 기록 지우기

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

다음 예에서는 비밀번호 기록을 지우고 트랜잭션을 커밋합니다.

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



6 장

이미지 관리

- 이미지 관리 정보, 45 페이지
- Cisco.com에서 이미지 다운로드, 46 페이지
- Firepower 9300 새시에 Firepower eXtensible 운영 체제 소프트웨어 이미지 다운로드, 46 페이지
- 이미지의 무결성 확인, 47 페이지
- Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드, 48 페이지
- 논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 9300 새시, 49 페이지
- 논리적 디바이스를 위한 이미지 버전 업데이트, 51 페이지
- 펌웨어 업그레이드, 53 페이지

이미지 관리 정보

Firepower 9300 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 Supervisor(관리자) 및 Firepower 보안 모듈/엔진에서 작동하는 여러 개별 이미지가 모여 있는 컬렉션입니다. 플랫폼 번들은 Firepower eXtensible 운영 체제 소프트웨어 패키지입니다.
- 애플리케이션 - 애플리케이션 이미지는 Firepower 9300 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 Supervisor(관리자)에 저장됩니다. 동일한 애플리케이션 이미지 유형의 서로 다른 여러 버전을 Firepower Supervisor(관리자)에 저장할 수 있습니다.



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

Cisco.com에서 이미지 다운로드

FXOS 및 애플리케이션 이미지를 Firepower 새시에 업로드할 수 있도록 Cisco.com에서 다운로드합니다.

시작하기 전에

Cisco.com 어카운트가 있어야 합니다.

프로시저

단계 1 웹 브라우저를 사용하여 <http://www.cisco.com/go/firepower9300-software> 또는 <http://www.cisco.com/go/firepower4100-software>로 이동합니다.

Firepower 9300 새시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.

단계 2 적절한 소프트웨어 이미지를 찾은 다음 로컬 컴퓨터에 다운로드합니다.

Firepower 9300 새시에 Firepower eXtensible 운영 체제 소프트웨어 이미지 다운로드

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

시작하기 전에

구성 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

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

프로시저

단계 1 펌웨어 모드를 입력합니다.

```
Firepower-chassis # scope firmware
```

단계 2 FXOS 소프트웨어 이미지를 다운로드합니다.

```
Firepower-chassis /firmware # download image URL
```

다음 구분 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
```

예

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

Firepower 9300 새시에 새 이미지가 추가되면 이미지의 무결성이 자동으로 확인됩니다. 필요한 경우 다음 절차를 사용하여 이미지의 무결성을 수동으로 확인할 수 있습니다.

프로시저

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

단계 2 펌웨어 모드를 입력합니다.

```
Firepower-chassis# scope firmware
```

단계 3 이미지를 나열합니다.

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

단계 4 이미지를 확인합니다.

```
Firepower-chassis /firmware # verify platform-pack version version_number
```

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

단계 5 확인하는 데 몇 분 정도 걸릴 수 있다는 메시지가 표시됩니다.

yes를 입력하여 검증을 계속할 것인지 확인합니다.

단계 6 이미지 확인 상태를 점검하려면:

```
Firepower-chassis /firmware # show validate-task
```

Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드

시작하기 전에

Cisco.com에서 플랫폼 번들 소프트웨어 이미지를 다운로드([Cisco.com에서 이미지 다운로드, 46 페이지 참조](#))한 다음 해당 이미지를 Firepower 9300 새시에 다운로드합니다([논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 9300 새시, 49 페이지 참조](#)).

프로시저

단계 1 FXOS CLI에 연결합니다([액세스 - FXOS CLI, 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.

yes를 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.

논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 9300 새시

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

시작하기 전에

구성 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

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

프로시저

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**

예

다음 예에서는 SCP 프로토콜을 사용하여 이미지를 복사합니다.

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

Downloads for Application Software:

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

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

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

Application:

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

Firepower-chassis /ssa # **scope app asa 9.4.1.65**

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

Application:

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

App Attribute Key for the Application:

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

Net Mgmt Bootstrap Key for the Application:

Bootstrap Key	Key Data	Type	Is the Key Secret	Description
PASSWORD	String	Yes		The admin user password.

Port Requirement for the Application:

```
Port Type: Data
```

```

Max Ports: 120
Min Ports: 1

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

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

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

논리적 디바이스를 위한 이미지 버전 업데이트

ASA 애플리케이션 이미지를 새 버전으로 업그레이드하거나 Firepower Threat Defense 애플리케이션 이미지를 재해 복구 시나리오에 사용할 새 시작 버전으로 설정하려면 다음 절차를 수행합니다.

ASA 논리적 디바이스에서 시작 버전을 변경하면 ASA가 해당 버전으로 업그레이드되며 모든 구성이 복원됩니다. 구성에 따라 ASA 시작 버전을 변경하려면 다음 워크플로를 사용합니다.

ASA 고가용성 -

1. 스탠바이 유닛에서 논리적 디바이스 이미지 버전을 변경합니다.
2. 스탠바이 유닛을 활성 상태로 설정합니다.
3. 다른 유닛에서 애플리케이션 버전을 변경합니다.

ASA 새시 간 클러스터 -

1. 슬레이브 유닛에서 시작 버전을 변경합니다.
2. 슬레이브 유닛을 마스터 유닛으로 설정합니다.
3. 원래 마스터 유닛(현재 슬레이브)에서 시작 버전을 변경합니다.

시작하기 전에

Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드([Cisco.com에서 이미지 다운로드, 46 페이지 참조](#))한 다음 해당 이미지를 Firepower 9300 새시에 다운로드합니다([논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 9300 새시, 49 페이지 참조](#)).

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

프로시저

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 설정합니다.

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

단계 5 구성을 커밋합니다.

```
commit-buffer
```

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

예

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

Firepower 9300 새시에서 펌웨어를 업그레이드하려면 다음 절차를 사용하십시오.

프로시저

-
- 단계 1** 웹 브라우저를 사용하여 <http://www.cisco.com/go/firepower9300-software> 또는 <http://www.cisco.com/go/firepower4100-software>로 이동합니다.
Firepower 9300 새시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.
- 단계 2** Cisco.com에서 적절한 펌웨어 패키지를 찾은 후 Firepower 9300 새시에서 액세스할 수 있는 서버로 다운로드합니다.
- 단계 3** Firepower 9300 새시에서 펌웨어 모드로 들어갑니다.
Firepower-chassis # **scope firmware**
- 단계 4** FXOS 펌웨어 이미지를 Firepower 9300 새시로 다운로드합니다.
Firepower-chassis /firmware # **download image URL**
다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.
- **ftp:// username@hostname / path**
 - **scp:// username@hostname / path**
 - **sftp:// username@hostname / path**
 - **tftp:// hostname : port-num / path**
- 단계 5** 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.
Firepower-chassis /firmware # **show download-task image_name detail**
- 단계 6** 다운로드가 완료되면 다음 명령을 입력하여 펌웨어 패키지의 내용을 볼 수 있습니다.
Firepower-chassis /firmware # **show package image_name expand**
- 단계 7** 다음 명령을 입력하여 펌웨어 패키지의 버전 번호를 볼 수 있습니다.
Firepower-chassis /firmware # **show package**
이 버전 번호는 펌웨어 패키지를 설치할 때 다음 단계에서 사용됩니다.
- 단계 8** 펌웨어 패키지를 설치하려면 다음과 같이 합니다.
- a) 펌웨어 설치 모드로 들어갑니다.
Firepower-chassis /firmware # **scope firmware-install**
 - b) 펌웨어 패키지를 설치합니다.
Firepower-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 Firepower 9300 새시의 전원을 껐다가 켜지 마십시오.

단계 9 업그레이드 프로세스를 모니터링하려면 다음과 같이 합니다.

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

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

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

```
Firepower-chassis # scope chassis 1
```

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

예

다음 예에서는 펌웨어 버전을 1.0.10으로 업그레이드합니다.

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

Download task:

```
File Name: fxos-k9-fpr9k-firmware.1.0.10.SPA
Protocol: Tftp
Server: 10.10.10.1
Port: 0
Userid:
```

```
Path:
```

```
Downloaded Image Size (KB): 2104
Time stamp: 2015-12-04T23:51:57.846
```

```
State: Downloading
```

```
Transfer Rate (KB/s): 263.000000
```

```
Current Task: unpacking image fxos-k9-fpr9k-firmware.1.0.10.SPA on primary(
```

```
FSM-STAGE:sam:dme:FirmwareDownloaderDownload:UnpackLocal)
```

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

```
Package fxos-k9-fpr9k-firmware.1.0.10.SPA:
```

```
Images:
```

```
fxos-k9-fpr9k-fpga.1.0.5.bin
fxos-k9-fpr9k-rommon.1.0.10.bin
```

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

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


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

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

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

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

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

Upgrading FXOS SUP firmware software package version 1.0.10

command executed
```




7 장

플랫폼 설정

- 관리 IP 주소 변경, 57 페이지
- 날짜 및 시간 설정, 59 페이지
- SSH 구성, 64 페이지
- 텔넷 구성, 65 페이지
- SNMP 구성, 66 페이지
- HTTPS 구성, 74 페이지
- AAA 구성, 87 페이지
- Syslog 구성, 98 페이지
- DNS 서버 구성, 100 페이지

관리 IP 주소 변경

시작하기 전에

Firepower 9300 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.

프로시저

단계 1 FXOS CLI에 연결합니다(액세스 - FXOS CLI, 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 구성에 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 구성의 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 구성에 트랜잭션을 커밋합니다.

```
Firepower-chassis /fabric-interconnect/ipv6-config* # commit-buffer
```

예

다음 예에서는 IPv4 관리 인터페이스 및 게이트웨이를 구성합니다.

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

```
Fabric Interconnect:
  ID   OOB IP Addr   OOB Gateway   OOB Netmask   OOB IPv6 Address OOB IPv6 Gateway
  Prefix Operability
-----
```

```
  A   192.0.2.112   192.0.2.1     255.255.255.0  ::              ::
  64   Operable
```

```
Firepower-chassis /fabric-interconnect # set out-of-band ip 192.0.2.111 netmask 255.255.255.0 gw 192.0.2.1
```

```
Warning: When committed, this change may disconnect the current CLI session
```

```
Firepower-chassis /fabric-interconnect* #commit-buffer
```

```
Firepower-chassis /fabric-interconnect #
```

다음 예에서는 IPv6 관리 인터페이스 및 게이트웨이를 구성합니다.

```

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

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

```

날짜 및 시간 설정

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

NTP 설정은 Firepower 9300 새시 및 새시에 설치된 논리적 디바이스 간에 자동으로 동기화됩니다.

표준 시간대 설정

프로시저

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
```

예

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.

프로시저

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 구성에 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 # scope ntp-server {hostname | ip-addr | ip6-addr}
```

```
Firepower-chassis /system/services/ntp-server # show detail
```

예

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

프로시저

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 구성에 트랜잭션을 커밋합니다.

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


예

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 클럭 수정사항은 즉시 적용됩니다.



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

프로시저

단계 1 시스템 모드를 입력합니다.

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

단계 2 시스템 서비스 모드를 입력합니다.

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

단계 3 시스템 클럭을 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 클럭 수정사항은 즉시 적용됩니다. 버퍼를 커밋할 필요가 없습니다.

예

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 방법과 SSH 클라이언트로 FXOS 새시를 활성화하는 방법을 설명합니다. SSH는 기본적으로 활성화되어 있습니다.

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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 구성은 현재 CLI를 사용하는 경우에만 사용할 수 있습니다.

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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 새시에서 SNMP(Simple Network Management Protocol)를 구성하는 방법을 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

SNMP 정보

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

SNMP 프레임워크는 다음 3가지 항목으로 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 관리자와 에이전트 간의 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 SNMPv2c는 모두 보안 커뮤니티 기반 양식을 사용합니다. SNMP는 다음에 정의되어 있습니다.

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

SNMP 알림

SNMP의 주요 기능은 SNMP 에이전트에서 알림을 생성하는 기능입니다. 이러한 알림에는 SNMP 관리자가 요청을 전송하지 않아도 됩니다. 알림은 잘못된 사용자 인증, 재시작, 연결 종료, 네이버 라우터에 대한 연결 손실 또는 기타 중요한 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 수준 및 권한

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

보안 수준은 SNMP 트랩에 연결된 메시지를 표시하는 데 필요한 권한을 결정합니다. 권한 수준은 메시지가 공개되지 않도록 보호해야 하는지 또는 인증되어야 하는지를 결정합니다. 어떤 보안 모델이 구현되는지에 따라 지원되는 보안 수준이 달라집니다. SNMP 보안 수준은 다음 권한 중 하나 이상을 지원합니다.

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

SNMPv3는 보안 모델 및 보안 수준을 모두 제공합니다. 보안 모델은 사용자 및 사용자 역할을 위해 설정된 인증 전략입니다. 보안 수준은 보안 모델에서 허용된 보안 수준입니다. 보안 모델과 보안 수준을 결합하여 SNMP 패킷을 처리할 때 어떤 보안 메커니즘이 적용되는지 결정합니다.

지원되는 SNMP 보안 모델과 수준 결합

다음 표에서는 어떻게 보안 모델과 수준을 결합할 수 있는지에 대해 설명합니다.

표 4: SNMP 보안 모델과 수준

모델	수준	인증	암호화	결과
v1	noAuthNoPriv	커뮤니티 문자열	없음	인증에 커뮤니티 문자열 일치를 사용합니다.
v2c	noAuthNoPriv	커뮤니티 문자열	없음	인증에 커뮤니티 문자열 일치를 사용합니다.
v3	noAuthNoPriv	Username	없음	인증에 사용자 이름 일치를 사용합니다.

모델	수준	인증	암호화	결과
v3	authNoPriv	HMAC-SHA	No(아니요)	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) 56 비트 암호화도 제공합니다.

SNMPv3 보안 기능

SNMPv3는 네트워크에서 인증 및 암호화 프레임워크를 결합하여 디바이스에 대한 보안 액세스를 제공합니다. SNMPv3는 구성된 사용자가 수행하는 관리 작업에만 권한을 부여하고 SNMP 메시지를 암호화합니다. SNMPv3 USM(User-Based Security Model)은 SNMP 메시지 수준 보안을 참조하며 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.

사용 가능한 MIB 및 MIB를 받을 수 있는 위치에 대한 내용은 [Cisco FXOS MIB 참조 가이드](#)를 참조하십시오.

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 구성을 활성화하고 SNMPv3 사용자에게 대한 프라이버시 비밀번호가

있는 경우, Firepower 새시는 해당 프라이버시 비밀번호를 사용하여 128비트 AES 키를 생성합니다. AES 프라이버시 비밀번호에는 최소 8자 이상을 포함할 수 있습니다. 암호가 일반 텍스트로 지정된 경우, 최대 64자를 지정할 수 있습니다.

SNMP 활성화 및 SNMP 속성 구성

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

다음의 예에서는 SNMP를 활성화하고 SNMP 커뮤니티 SnpCommSystem2를 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: SnpCommSystem2
```

```
Firepower-chassis /monitoring* # set snmp syscontact contactperson1
Firepower-chassis /monitoring* # set snmp syslocation systemlocation
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

다음에 수행할 작업

SNMP 트랩 및 사용자를 생성합니다.

SNMP 트랩 생성

프로시저

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}
```

다음을 선택할 수 있습니다.

- 버전으로 v2c 또는 v3를 선택한 경우 **traps**
- 버전으로 v2c를 선택한 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}
```

다음을 선택할 수 있습니다.

- **auth** — 인증하지만 암호화 없음
- **noauth** — 인증 또는 암호화 없음
- **priv** — 인증 및 암호화

단계 9 시스템 구성에 트랜잭션을 커밋합니다.

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

예

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

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

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

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

SNMP 트랩 삭제

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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

프로시저

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.

Firepower eXtensible 운영 체제에서는 다음 요건을 충족하지 않는 모든 비밀번호를 거부합니다.

- 8자 이상, 80자 이하여야 합니다.
- 문자, 숫자 및 다음 문자만 포함해야 합니다.
~!@#%^&*()-+{}[]\|:;'"<>./
- 다음 기호는 포함할 수 없습니다. 예: \$(달러 기호), ? (물음표) 또는 = (등호).
- 서로 다른 문자를 5개 이상 포함해야 합니다.
- 연속적으로 증가하거나 감소하는 문자나 숫자를 너무 많이 포함하면 안 됩니다. 예를 들어 "12345" 문자열에는 이러한 문자가 4개 포함되고 "ZYXW" 문자열에는 3개 포함됩니다. 이러한 문자의 총 수가 특정 한도를 초과하는 경우(대개 해당 문자가 4~6개 이상 포함되는 경우) 단순성 검사에서 장애가 발생합니다.

참고 연속적으로 증가하거나 감소하는 문자 사이에 증가하거나 감소하지 않는 문자가 사용되는 경우에는 증가/감소 문자 수가 재설정되지 않습니다. 예를 들어 abcd&!21의 경우 비밀번호 검사에서 장애가 발생하지만 abcd&!25의 경우에는 장애가 발생하지 않습니다.

단계 4 AES-128 암호화 사용을 활성화 또는 비활성화합니다.

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

기본적으로 AES-128 암호화는 비활성화되어 있습니다.

단계 5 사용자 프라이버시 비밀번호를 지정합니다.

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

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

Firepower eXtensible 운영 체제에서는 다음 요건을 충족하지 않는 모든 비밀번호를 거부합니다.

- 8자 이상, 80자 이하여야 합니다.
- 문자, 숫자 및 다음 문자만 포함해야 합니다.
~!@#%&*()_+{}[]\|:;'"<>./
- 다음 기호는 포함할 수 없습니다. 예: \$(달러 기호), ? (물음표) 또는 = (등호).
- 서로 다른 문자를 5개 이상 포함해야 합니다.
- 연속적으로 증가하거나 감소하는 문자나 숫자를 너무 많이 포함하면 안 됩니다. 예를 들어 "12345" 문자열에는 이러한 문자가 4개 포함되고 "ZYXW" 문자열에는 3개 포함됩니다. 이러한 문자의 총 수가 특정 한도를 초과하는 경우(대개 해당 문자가 4~6개 이상 포함되는 경우) 단순성 검사에서 장애가 발생합니다.

참고 연속적으로 증가하거나 감소하는 문자 사이에 증가하거나 감소하지 않는 문자가 사용되는 경우에는 증가/감소 문자 수가 재설정되지 않습니다. 예를 들어 abcd&!21의 경우 비밀번호 검사에서 장애가 발생하지만 abcd&!25의 경우에는 장애가 발생하지 않습니다.

단계 6 시스템 구성에 트랜잭션을 커밋합니다.

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

예

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

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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

```

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

```

HTTPS 구성

이 섹션에서는 Firepower 9300 새시에서 SNMP를 구성하는 방법을 설명합니다.



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

인증서, 키 링, 트러스트 포인트

HTTPS에서는 PKI(Public Key Infrastructure)의 구성 요소를 사용하여 두 디바이스, 이를테면 클라이언트 브라우저와 Firepower 9300 새시 간의 보안 통신을 설정합니다.

암호화 키 및 키 링

각 PKI 디바이스는 비대칭 RSA(Rivest-Shamir-Adleman) 암호화 키의 쌍을 보유합니다. 개인 키와 공개 키로 구성된 이 쌍은 내부 키 링에 저장됩니다. 두 키 중 하나로 암호화된 메시지는 나머지 키로 해독할 수 있습니다. 암호화된 메시지를 보낼 때 발신자는 수신자의 공개 키로 메시지를 암호화하며 수신자는 자신의 개인 키로 그 메시지를 해독합니다. 또한 발신자는 자체 개인 키로 알려진 메시지를 암호화('서명'이라고도 함)하여 공개 키의 소유권을 증명할 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. 일반적으로는 길이가 더 긴 키가 짧은 키보다 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 FXOS에는 기본 키 링의 공개 키를 포함하는 자체 서명 인증서가 내장되어 있습니다.

신뢰 지점

FXOS에 대한 더 강력한 인증을 제공하기 위해 신뢰할 수 있는 출처 또는 트러스트 포인트로부터 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 가져오려면 FXOS를 통해 인증서 요청을 생성하여 트러스트 포인트에 제출해야 합니다.



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

키 링 생성

FXOS는 기본 키 링을 포함하여 최대 8개의 키 링을 지원합니다.

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 키 링의 이름을 생성합니다.

```
Firepower-chassis # create keyring keyring-name
```

단계 3 SSL 키 길이(비트)를 설정합니다.

```
Firepower-chassis # set modulus {mod1024 | mod1536 | mod2048 | mod512}
```

단계 4 트랜잭션을 커밋합니다.

```
Firepower-chassis # commit-buffer
```

예

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 수행할 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.

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 기본 키 링에 대한 키 링 보안 모드로 들어갑니다.

```
Firepower-chassis /security # scope keyring default
```

단계 3 기본 키 링 재생성:

```
Firepower-chassis /security/keyring # set regenerate yes
```

단계 4 트랜잭션을 커밋합니다.

```
Firepower-chassis # commit-buffer
```

예

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

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 키 링에 대한 구성 모드로 들어갑니다.

```
Firepower-chassis /security # scope keyring keyring-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-name name}
```

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
```

예

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-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYWljMDQwZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF80PhKbhghLAlYZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHO4SwccAUXQ5Zngf45YtXlWsyUUV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBqkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQMA6CBnNhbWwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAQBQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHU003Teg
nhsyu4satpyiPqVv9viKZ+spvc6x5PWIcTWgHhH8BimOb/00KuG8kwfIGGSd1Av
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGx1DNqoN+odCXPc5kjoXD01ZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring #

```

다음에 수행할 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

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 키 링에 대한 구성 모드로 들어갑니다.

```
Firepower-chassis /security # scope keyring keyring-name
```

단계 3 인증서 요청을 생성합니다.

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

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 Firepower 9300 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 Firepower 9300 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
```

예

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 bg1-samc-15A
Firepower-chassis /security/keyring/certreq* # set email test@cisco.com
Firepower-chassis /security/keyring/certreq* # set locality new york city
Firepower-chassis /security/keyring/certreq* # set org-name "Cisco Systems"
```

```

Firepower-chassis /security/keyring/certreq* # set org-unit-name Testing
Firepower-chassis /security/keyring/certreq* # set state new york
Firepower-chassis /security/keyring/certreq* # commit-buffer
Firepower-chassis /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name: test@cisco.com
Certificate request country name: US
State, province or county (full name): New York
Locality name (eg, city): new york city
Organization name (eg, company): Cisco
Organization Unit name (eg, section): Testing
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYWljMDQwZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF80PhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHO4SwccAUXQ5Zngf45YtX1WsywUWV4
0re/zgTk/WCd56RFOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQMA6CBnNhbWwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCszN0qUHYGFoQw56RwQueLTNPnrndqUwuZHU003Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWICtWgHhH8BimOb/00KuG8kwfIGGSed1Av
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGx1DNqon+odCXpc5kjoXD01ZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring/certreq #

```

다음에 수행할 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.

트러스트 포인트 생성

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 신뢰 지점을 생성합니다.

```
Firepower-chassis /security # create trustpoint name
```

단계 3 이 신뢰 지점에 대한 인증서 정보를 지정합니다.

```
Firepower-chassis /security/trustpoint # set certchain [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.

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

단계 4 트랜잭션을 커밋합니다.

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

예

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-----
> MIIDMDCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGx1IEluYy4xEzARBGNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWZVZlZmVzZXJAZXhhbXBsZS5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMiVvCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgd4VBNKOND1
> GmbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bf5wZVNAgMBAAGGJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzC190306Mg51zq1zXcz75+VFj2I6rH9asckClD3mkOVx5gJU
> Ptt5CVQpNgNldvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtv1WvfhevskV0j6
> jtceMyZ+f7+3yh42lido3n04MIGeBgNVHSMegZYwgZOAF1NjtcEMyZ+f7+3yh42
> lido3n04oXikdjb0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVAcT
> ClNhbnRhIENsYXJhMRswGQYDVQQKEwJ0dW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0VvZ21uZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAwADAyDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WWvB5fKqGQXc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8cop1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIzJwK5+N3x/kfa2EHU6id1avt
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/trustpoint* # commit-buffer
Firepower-chassis /security/trustpoint #
```

다음에 수행할 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.

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 인증서를 수신할 키 링에 대한 구성 모드로 들어갑니다.

```
Firepower-chassis /security # scope keyring keyring-name
```

단계 3 키 링 인증서를 수신한 Trust anchor 또는 인증 증명에 대한 신뢰 지점을 지정합니다.

```
Firepower-chassis /security/keyring # set trustpoint name
```

단계 4 키 링 인증서를 입력 및 업로드할 대화 상자를 엽니다.

```
Firepower-chassis /security/keyring # set cert
```

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

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

단계 5 트랜잭션을 커밋합니다.

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

예

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/zCCAAGCAQAwgZkxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTEVMBMGAlUE
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGx1IEluYy4xEzARBgNVBAST
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWElVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBKOND1
> GMbkPayV1Qjbg4MD2dx2+H8EH3LMtdZrgKvPxPTE+bf5wZVNAGMBAAGgJTAjBqkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVmhZCl90306Mg51zq1zXcz75+VFj2I6rH9asckClD3mkOVx5gJU
> Ptt5CVQpNgNldvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtv1WvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L4317IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 수행할 작업
HTTPS 서비스를 키 링으로 구성합니다.

HTTPS 구성



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

프로시저

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-suite를 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 설정되어 있으면 이 옵션은 무시됩니다.

단계 8 시스템 구성에 트랜잭션을 커밋합니다.

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

예

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.

프로시저

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 구성에 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 구성을 설정하는 동안 입력한 Firepower 새시의 IP 주소 또는 호스트 이름이며 `<chassis_mgmt_port>`는 방금 구성한 HTTPS 포트입니다.

예

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

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 명명된 키 링을 삭제합니다.

```
Firepower-chassis /security # delete keyring name
```

단계 3 트랜잭션을 커밋합니다.

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

예

다음 예에서는 사용자 계정을 삭제합니다.

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

트러스트 포인트 삭제

시작하기 전에

신뢰 지점이 키 링에서 사용하지 않음을 확인합니다.

프로시저

단계 1 보안 모드로 들어갑니다.

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

단계 2 명명된 신뢰 지점을 삭제합니다.

```
Firepower-chassis /security # delete trustpoint name
```

단계 3 트랜잭션을 커밋합니다.

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

예

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

프로시저

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 구성에 트랜잭션을 커밋합니다.


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

예

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 크리덴셜을 데이터베이스에 저장된 다른 사용자의 크리덴셜과 비교합니다. 크리덴셜이 일치하면 사용자는 네트워크에 액세스할 수 있습니다. 크리덴셜이 일치하지 않으면, 인증에 실패하고 네트워크 액세스가 거부됩니다.

다음 세션을 포함하는 새시에 대한 관리 연결을 인증하도록 Firepower 9300 새시를 구성할 수 있습니다.

- HTTPS
- SSH
- 시리얼 콘솔

권한 부여

권한 부여는 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.

어카운팅

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 권한 부여 제어, 청구, 경향 분석, 리소스 활용도 및 용량 계획 활동에 사용됩니다.

인증, 권한 부여 및 어카운팅 간 상호 작용

인증을 단독으로 사용하거나 권한 부여 및 어카운팅과 함께 사용할 수 있습니다. 권한 부여에서는 항상 먼저 사용자의 인증 여부를 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. 권한 부여는 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 사업자에 이러한 속성 중 하나에 대한 설정이 포함된 경우 Firepower eXtensible 운영 체제에서는 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.

프로시저

단계 1 보안 모드를 입력합니다.

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

단계 2 보안 LDAP 모드를 입력합니다.

```
Firepower-chassis /security # scope ldap
```

단계 3 지정된 속성을 포함하는 레코드로 데이터베이스 검색을 제한합니다.

```
Firepower-chassis /security/ldap # set attribute attribute
```

단계 4 지정된 고유 이름을 포함하는 레코드로 데이터베이스 검색을 제한합니다.

```
Firepower-chassis /security/ldap # set basedn distinguished-name
```

단계 5 지정된 필터를 포함하는 레코드로 데이터베이스 검색을 제한합니다.

```
Firepower-chassis /security/ldap # set filter filter
```

단계 6 서버가 다운되었다고 인지할 때까지 시스템이 LDAP 서버의 응답을 대기해야 하는 시간 간격을 설정합니다.

```
Firepower-chassis /security/ldap # set timeout seconds
```

단계 7 시스템 구성에 트랜잭션을 커밋합니다.

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

예

다음 예에서는 LDAP 속성을 CiscoAvPair로, 기본 고유 이름을 "DC=cisco-firepower-aaa3,DC=qalab,DC=com"으로, 필터를 sAMAccountName=\$userid로, 시간 초과 간격을 5초로 각각 설정하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap # set attribute CiscoAvPair
Firepower-chassis /security/ldap* # set basedn "DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap* # set filter sAMAccountName=$userid
Firepower-chassis /security/ldap* # set timeout 5
Firepower-chassis /security/ldap* # commit-buffer
Firepower-chassis /security/ldap #
```



참고 사용자 로그인은 LDAP 사용자의 userdn이 255자를 초과하는 경우 실패합니다.

다음에 수행할 작업

LDAP 제공자를 생성합니다.

LDAP 제공자 생성

Firepower eXtensible 운영 체제에서는 최대 16개의 LDAP 사업자를 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.

프로시저

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 속성을 설정합니다.

```
Firepower-chassis /security/ldap/server # set attribute attr-name
```

이 속성은 항상 이름값 쌍입니다. 시스템은 사용자 레코드를 쿼리하여 이 속성 이름과 일치하는 값을 찾습니다.

이 값은 기본 속성이 LDAP 제공자에 대해 설정되지 않은 경우 필요합니다.

단계 5 (선택 사항) 원격 사용자가 로그인하고 시스템에서 사용자 이름에 기초한 사용자 DN을 얻으려고 시도할 때 서버에서 검색을 시도해야 하는 LDAP 계층 구조에서 특정한 고유 이름을 설정합니다.

```
Firepower-chassis /security/ldap/server # set basedn basedn-name
```

기본 DN 길이는 최대 255자에서 CN=\$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) 또는 =(등호)를 제외한 모든 표준 ASCII 문자를 입력할 수 있습니다.

비밀번호를 설정하려면 **set password** 명령을 입력한 후 **Enter** 키를 누르고 프롬프트에 키 값을 입력합니다.

단계 9 (선택 사항) Firepower eXtensible 운영 체제에서 사용자를 인증하기 위해 이 사업자를 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.

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

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

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

1~60초의 정수를 입력하거나 0(숫자 0)을 입력하여 LDAP 제공자에 지정된 전역 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 시스템 구성에 트랜잭션을 커밋합니다.

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

예

다음의 예에서는 10.193.169.246이라는 이름의 LDAP 서버 인스턴스를 생성하고 bind, 비밀번호, 순서, 포트, SSL 설정, 벤더 속성을 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap* # create server 10.193.169.246
```

```

Firepower-chassis /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap/server* # set password
Enter the password:
Confirm the password:
Firepower-chassis /security/ldap/server* # set order 2
Firepower-chassis /security/ldap/server* # set port 389
Firepower-chassis /security/ldap/server* # set ssl yes
Firepower-chassis /security/ldap/server* # set timeout 30
Firepower-chassis /security/ldap/server* # set vendor ms-ad
Firepower-chassis /security/ldap/server* # commit-buffer
Firepower-chassis /security/ldap/server #

```

다음의 예에서는 12:31:71:1231:45b1:0011:011:900이라는 이름의 LDAP 서버 인스턴스를 생성하고 binddn, 비밀번호, 순서, 포트, SSL 설정, 벤더 속성을 구성하고 트랜잭션을 커밋합니다.

```

Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap* # create server 12:31:71:1231:45b1:0011:011:900
Firepower-chassis /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap/server* # set password
Enter the password:
Confirm the password:
Firepower-chassis /security/ldap/server* # set order 1
Firepower-chassis /security/ldap/server* # set port 389
Firepower-chassis /security/ldap/server* # set ssl yes
Firepower-chassis /security/ldap/server* # set timeout 45
Firepower-chassis /security/ldap/server* # set vendor ms-ad
Firepower-chassis /security/ldap/server* # commit-buffer
Firepower-chassis /security/ldap/server #

```

LDAP 제공자 삭제

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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 사업자에 이러한 속성 중 하나에 대한 설정이 포함된 경우 Firepower eXtensible 운영 체제에서는 해당 설정을 사용하고 기본 설정을 무시합니다.

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

다음의 예에서는 RADIUS 재시도 횟수를 4로 설정하고 시간 초과 간격을 30초로 설정하며 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
```

```
Firepower-chassis /security/radius # set retries 4
Firepower-chassis /security/radius* # set timeout 30
Firepower-chassis /security/radius* # commit-buffer
Firepower-chassis /security/radius #
```

다음에 수행할 작업

RADIUS 제공자를 생성합니다.

RADIUS 제공자 생성

Firepower eXtensible 운영 체제에서는 최대 16개의 RADIUS 제공자를 지원합니다.

프로시저

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 key** 명령을 입력한 후 **Enter** 키를 누르고 프롬프트에 키 값을 입력합니다.

단계 6 (선택 사항) 이 서버에 시도할 순서를 지정합니다.

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

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

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

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

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

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

단계 9 시스템 구성에 트랜잭션을 커밋합니다.

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

예

다음 예에서는 `radiusserv7`이라는 이름의 서버 인스턴스를 생성하고 인증 포트를 5858로 설정하고 키를 `radiuskey321`로 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

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

다음 예에서는 `radius1`이라는 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 사업자에 이러한 속성 중 하나에 대한 설정이 포함된 경우 Firepower eXtensible 운영 체제에서는 해당 설정을 사용하고 기본 설정을 무시합니다.

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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 수행할 작업

TACACS+ 제공자를 만듭니다.

TACACS+ 제공자 생성

Firepower eXtensible 운영 체제에서는 최대 16개의 TACACS+ 제공자를 지원합니다.

프로시저

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 key** 명령을 입력한 후 **Enter** 키를 누르고 프롬프트에 키 값을 입력합니다.

단계 5 (선택 사항) 이 서버에 시도할 순서를 지정합니다.

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

단계 6 서버가 다운되었다고 인지할 때까지 시스템이 TACACS+ 서버의 응답을 대기해야 하는 시간 간격을 지정합니다.

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

팁 TACACS+ 제공자에 대한 2단계 인증을 선택한 경우, 더 높은 Timeout(시간 초과) 값을 구성하는 것이 좋습니다.

단계 7 (선택 사항) TACACS+ 서버와의 통신에 사용되는 포트를 지정합니다.

```
Firepower-chassis /security/tacacs/server # set port port-num
```

단계 8 시스템 구성에 트랜잭션을 커밋합니다.

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

예

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

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

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 구성 파일에 따라 인쇄합니다. 이 로깅 양식을 통해 로그를 안전하게 장기 보관할 수 있습니다. 로그는 일상적인 트러블슈팅과 사고 처리에 모두 유용합니다.

프로시저

단계 1 모니터링 모드를 입력합니다.

```
Firepower-chassis# scope monitoring
```

단계 2 콘솔로의 syslogs 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
```

예

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 명령에 DNS 서버 조치가 필요한 경우, 임의 순서로 DNS 서버 3개만 검색할 수 있습니다.

프로시저

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 구성에 트랜잭션을 커밋합니다.

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

예

다음 예에서는 IPv4 주소 192.168.200.105를 사용하는 DNS 서버를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create dns 192.168.200.105
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

다음 예에서는 IPv6 주소 2001:db8::22:F376:FF3B:AB3F를 사용하는 DNS 서버를 구성하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create dns 2001:db8::22:F376:FF3B:AB3F
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

다음 예에서는 IP 주소 192.168.200.105를 사용하는 DNS 서버를 삭제하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete dns 192.168.200.105
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```




8 장

인터페이스 관리

- Firepower 인터페이스 정보, 103 페이지
- Firepower 인터페이스에 대한 지침 및 제한 사항, 104 페이지
- 인터페이스 구성, 105 페이지
- 모니터링 인터페이스, 112 페이지

Firepower 인터페이스 정보

Firepower 9300 새시에서는 물리적 인터페이스 및 EtherChannel(포트-채널) 인터페이스를 지원합니다. EtherChannel 인터페이스는 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

새시 관리 인터페이스

새시 관리 인터페이스는 SSH 또는 Firepower Chassis Manager에서 FXOS 새시 관리에 사용됩니다. 이 인터페이스는 애플리케이션 관리용 논리적 디바이스에 할당하는 관리 유형 인터페이스와는 별개입니다.

이 인터페이스의 파라미터는 CLI에서 구성해야 합니다. [관리 IP 주소 변경, 57 페이지](#) 섹션도 참조하십시오. FXOS CLI에서 이 인터페이스에 대한 정보를 확인하려면 로컬 관리에 연결한 다음 관리 포트를 표시합니다.

```
Firepower # connect local-mgmt
```

```
Firepower(local-mgmt) # show mgmt-port
```

실제 케이블이나 SFP 모듈 연결을 해제하거나 **mgmt-port shut** 명령을 수행하더라도 새시 관리 인터페이스는 계속 작동합니다.

인터페이스 유형

각 인터페이스는 다음 유형 중 하나일 수 있습니다.

- Data(데이터) - 데이터 인터페이스는 논리적 디바이스 간에 공유할 수 없습니다.

- **Mgmt(관리)** - 관리 인터페이스를 사용하여 애플리케이션 인스턴스를 관리합니다. 하나 이상의 논리적 디바이스가 외부 호스트에 액세스하기 위해 이러한 인터페이스를 공유할 수 있습니다. 논리적 디바이스가 인터페이스를 공유하는 다른 논리적 디바이스와 이 인터페이스를 통해 통신할 수는 없습니다. 논리적 디바이스당 관리 인터페이스 1개만 할당할 수 있습니다. 개별 새시 관리 인터페이스에 대한 내용은 [새시 관리 인터페이스, 103 페이지](#) 섹션을 참조하십시오.
- **Cluster(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.

새시와 애플리케이션의 독립 인터페이스 상태

관리를 위해 새시와 애플리케이션에서 인터페이스를 활성화하고 비활성화할 수 있습니다. 인터페이스는 두 운영 체제에서 모두 활성화해야 작동합니다. 인터페이스 상태는 독립적으로 제어되므로 새시와 애플리케이션에서 상태가 일치하지 않을 수도 있습니다.

Jumbo Frame Support

Firepower 9300 새시에서는 기본적으로 점보 프레임 지원이 활성화되어 있습니다. Firepower 9300 새시에 설치된 특정 논리적 디바이스에서 점보 프레임 지원을 활성화하려면 논리적 디바이스에서 인터페이스에 대한 적절한 MTU 설정을 구성해야 합니다.

Firepower 9300 새시의 애플리케이션에 대해 지원되는 최대 MTU는 9000입니다.

Firepower 인터페이스에 대한 지침 및 제한 사항

인라인 집합 **FTD**

- 물리적 인터페이스 전용(일반 포트와 breakout 포트 둘 다)으로 지원되며 EtherChannel은 지원되지 않습니다.
- 링크 상태 전파가 지원되지 않습니다.

기본 **MAC** 주소

기본 MAC 주소 할당은 인터페이스의 유형에 따라 다릅니다.

- 물리적 인터페이스 - 물리적 인터페이스는 버닝된 MAC 주소를 사용합니다.
- EtherChannel - EtherChannel의 경우 채널 그룹에 속한 모든 인터페이스가 동일한 MAC 주소를 공유합니다. 이 기능은 EtherChannel을 네트워크 애플리케이션 및 사용자에게 투명하게 만듭니다. 이들은 논리적 연결만 볼 수 있으며, 개별 링크에 대해서는 모르기 때문입니다. 포트 채널 인터페이스는 풀의 고유 MAC 주소를 사용하며 인터페이스 멤버십은 MAC 주소에 영향을 주지 않습니다.

인터페이스 구성

기본적으로 물리적 인터페이스는 비활성화되어 있습니다. 인터페이스를 활성화하고, EtherChannel 을 추가하고, 인터페이스 속성을 수정하고, breakout 포트를 구성할 수 있습니다.

실제 인터페이스 구성

인터페이스를 물리적으로 활성화 및 비활성화할 뿐만 아니라 인터페이스 속도 및 듀플렉스를 설정할 수 있습니다. 인터페이스를 사용하려면 FXOS에서 인터페이스를 물리적으로 활성화하고 애플리케이션에서 논리적으로 활성화해야 합니다.

시작하기 전에

- 이미 EtherChannel의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. EtherChannel에 인터페이스를 추가하기 전에 설정을 구성하십시오.

프로시저

단계 1 인터페이스 모드를 설정합니다.

scope eth-uplink

scope fabric a

단계 2 인터페이스를 활성화합니다.

enter interface interface_id

enable

예제:

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

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

단계 3 (선택 사항) 인터페이스 유형을 설정합니다.

set port-type {data | mgmt | cluster}

예제:

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

data 키워드는 기본 유형입니다. **cluster** 키워드는 선택하지 마십시오. 기본적으로 클러스터 제어 링크는 Port-channel 48에서 자동으로 생성됩니다.

단계 4 인터페이스에 대해 지원되는 경우 자동 협상을 활성화하거나 비활성화합니다.

set auto-negotiation {on | off}

예제:

```
Firepower /eth-uplink/fabric/interface* # set auto-negotiation off
```

단계 5 인터페이스 속도를 설정합니다.

set admin-speed {10mbps | 100mbps | 1gbps | 10gbps | 40gbps | 100gbps}

예제:

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

단계 6 인터페이스 듀플렉스 모드를 설정합니다.

set admin-duplex {fullduplex | halfduplex}

예제:

```
Firepower /eth-uplink/fabric/interface* # set admin-duplex halfduplex
```

단계 7 기본 플로우 제어 정책을 수정한 경우 인터페이스에 정책이 이미 적용되어 있습니다. 새 정책을 생성한 경우에는 인터페이스에 정책을 적용합니다. [플로우 제어 정책 구성, 110 페이지](#) 섹션을 참조하십시오.

set flow-control-policy name

예제:

```
Firepower /eth-uplink/fabric/interface* # set flow-control-policy flow1
```

단계 8 구성을 저장합니다.

commit-buffer

예제:

```
Firepower /eth-uplink/fabric/interface* # commit-buffer
Firepower /eth-uplink/fabric/interface #
```

EtherChannel(포트 채널) 추가

EtherChannel(포트 채널)은 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다. LACP(Link Aggregation Control Protocol)에서는 두 네트워크 디바이스 간의 LACPDU(Link Aggregation Control Protocol Data Units)를 교환하여 인터페이스를 취합합니다.

Firepower 9300 새시는 각 멤버 인터페이스가 LACP 업데이트를 송수신할 수 있도록 액티브 LACP 모드에서만 EtherChannel을 지원합니다. 액티브 EtherChannel은 액티브 또는 패시브 EtherChannel과의 연결을 설정할 수 있습니다. LACP 트래픽 양을 최소화할 필요가 없는 한 액티브 모드를 사용해야 합니다.

LACP에서는 사용자의 작업 없이 EtherChannel에 링크를 자동으로 추가 및 삭제하는 작업을 조율합니다. 또한 구성 오류를 처리하고 멤버 인터페이스의 양끝이 모두 올바른 채널 그룹에 연결되어 있는지 확인합니다.

Firepower 9300 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 인터페이스 또는 관리 인터페이스로 추가됩니다.
- EtherChannel이 클러스터의 일부인 논리적 디바이스에 대한 관리 인터페이스 또는 클러스터 제어 링크로 추가됩니다.
- EtherChannel이 클러스터의 일부이며 유닛 하나 이상이 클러스터에 조인된 논리적 디바이스에 대한 데이터 인터페이스로 추가됩니다.

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

프로시저

단계 1 인터페이스 모드를 입력합니다.

```
scope eth-uplink
```

```
scope fabric a
```

단계 2 포트 채널을 생성합니다.

```
create port-channel id
```

```
enable
```

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 data
```

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 {10mbps | 100mbps | 1gbps | 10gbps | 40gbps | 100gbps}
```

예제:

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

단계 6 (선택 사항) 포트 채널의 모든 멤버에 대해 듀플렉스를 설정합니다.

```
set duplex {fullduplex | halfduplex}
```

예제:

```
Firepower /eth-uplink/fabric/port-channel* # set duplex fullduplex
```

단계 7 인터페이스에 대해 지원되는 경우 자동 협상을 활성화하거나 비활성화합니다.

```
set auto-negotiation {on | off}
```

예제:

```
Firepower /eth-uplink/fabric/interface* # set auto-negotiation off
```

단계 8 기본 플로우 제어 정책을 수정한 경우 인터페이스에 정책이 이미 적용되어 있습니다. 새 정책을 생성한 경우에는 인터페이스에 정책을 적용합니다. [플로우 제어 정책 구성, 110 페이지](#) 섹션을 참조하십시오.

```
set flow-control-policy name
```

예제:

```
Firepower /eth-uplink/fabric/interface* # set flow-control-policy flow1
```

단계 9 구성을 커밋합니다.

commit-buffer

분할 케이블 구성

다음 절차에서는 Firepower 9300 새시에서 사용할 분할 케이블을 구성하는 방법을 보여줍니다. 분할 케이블을 사용하여 40Gbps 포트 1개 대신 10Gbps 포트 4개를 제공할 수 있습니다.

프로시저

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) 구성을 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 Breakout 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 없음을 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) 구성을 커밋합니다.

commit-buffer

플로우 제어 정책 구성

플로우 제어 정책은 포트의 수신 버퍼가 찼을 때 이더넷 포트가 IEEE 802.3x 일시 중지 프레임을 보내고 받을지 여부를 결정합니다. 이 일시 중지 프레임은 버퍼가 비워질 때까지 몇 밀리초 동안 전송 포트에서 데이터 전송을 정지하도록 요청합니다. 디바이스 간에 플로우 제어가 이루어지려면 양쪽 디바이스 모두에서 수신 및 전송 플로우 제어 파라미터를 활성화해야 합니다.

기본 정책은 전송 및 수신 제어를 비활성화하며 우선 순위를 자동 협상으로 설정합니다.

프로시저

단계 1 eth-uplink 모드와 flow-control 모드를 차례로 설정합니다.

scope eth-uplink

scope flow-control

예제:

```
firepower-4110# scope eth-uplink
firepower-4110 /eth-uplink # scope flow-control
firepower-4110 /eth-uplink/flow-control #
```

단계 2 플로우 제어 정책을 수정하거나 생성합니다.

enter policy name

기본 정책을 수정하려면 이름으로 **default**를 입력합니다.

예제:

```
firepower-4110 /eth-uplink/flow-control # enter policy default
firepower-4110 /eth-uplink/flow-control/policy* #
```

단계 3 우선 순위를 설정합니다.

set prio {auto | on}

우선 순위에 따라 이 링크에 대해 PPP를 활성화할지 아니면 협상할지가 설정됩니다.

예제:

```
firepower-4110 /eth-uplink/flow-control/policy* # set prio on
```

단계 4 플로우 제어 수신 일시 중지를 활성화하거나 비활성화합니다.

set receive {on | off}

- **on(켜기)** - 일시 중지 요청을 수용하고, 네트워크에서 일시 중지 요청을 취소할 때까지 해당 업링크 포트에서 모든 트래픽을 중지합니다.
- **off(끄기)** - 네트워크의 일시 중지 요청을 무시하고 트래픽 플로우가 평소대로 진행됩니다.

예제:

```
firepower-4110 /eth-uplink/flow-control/policy* # set receive on
```

단계 5 플로우 제어 전송 일시 중지를 활성화하거나 비활성화합니다.

set send {on | off}

- **on(켜기)** - 수신 패킷 속도가 너무 높아지면 Firepower 9300에서 네트워크에 일시 중지를 요청합니다. 트래픽이 정상 레벨로 돌아올 때까지 몇 밀리초 동안 일시 중지됩니다.
- **off(끄기)** - 패킷 로드와 상관없이 포트 트래픽이 정상적으로 흐릅니다.

예제:

```
firepower-4110 /eth-uplink/flow-control/policy* # set send on
```

단계 6 구성을 저장합니다.

commit-buffer

예제:

```
firepower-4110 /eth-uplink/flow-control/policy* # commit-buffer
firepower-4110 /eth-uplink/flow-control/policy #
```

예

다음 예시에서는 플로우 제어 정책을 구성합니다.

```
firepower-4110# scope eth-uplink
firepower-4110 /eth-uplink # scope flow-control
firepower-4110 /eth-uplink/flow-control # enter policy FlowControlPolicy23
firepower-4110 /eth-uplink/flow-control/policy* # set prio auto
firepower-4110 /eth-uplink/flow-control/policy* # set receive on
firepower-4110 /eth-uplink/flow-control/policy* # set send on
firepower-4110 /eth-uplink/flow-control/policy* # commit-buffer
firepower-4110 /eth-uplink/flow-control/policy #
```

모니터링 인터페이스

• **show interface**

인터페이스 상태를 표시합니다.



참고 포트 채널에서 포트 역할을 하는 인터페이스는 이 목록에 나타나지 않습니다.

```
Firepower# scope eth-uplink
Firepower /eth-uplink # scope fabric a
Firepower /eth-uplink/fabric # show interface
```

Interface:

Port Name	Port Type	Admin State	Oper State	State Reason
Ethernet1/1	Mgmt	Enabled	Up	
Ethernet1/2	Data	Enabled	Link Down	Link failure or not-connected
Ethernet1/3	Data	Enabled	Up	
Ethernet1/4	Data	Enabled	Sfp Not Present	Unknown
Ethernet1/6	Data	Enabled	Sfp Not Present	Unknown
Ethernet1/7	Data	Enabled	Sfp Not Present	Unknown
Ethernet1/8	Data	Disabled	Sfp Not Present	Unknown
Ethernet2/1	Data	Enabled	Up	
Ethernet2/2	Data	Enabled	Up	
Ethernet2/4	Data	Enabled	Up	
Ethernet2/5	Data	Enabled	Up	
Ethernet2/6	Data	Enabled	Up	
Ethernet3/2	Data	Enabled	Up	
Ethernet3/4	Data	Enabled	Up	



9 장

논리적 디바이스

- 논리적 디바이스 정보, 113 페이지
- 논리적 디바이스의 요구 사항 및 사전 요구 사항, 114 페이지
- 논리적 디바이스 관련 지침 및 제한 사항, 115 페이지
- 독립형 논리적 디바이스 추가, 120 페이지
- 고가용성 쌍 추가, 125 페이지
- 클러스터 추가, 126 페이지
- 논리적 디바이스 관리, 138 페이지
- 논리적 디바이스 모니터링, 145 페이지
- 사이트 간 클러스터링 예시, 146 페이지
- 논리적 디바이스의 기록, 148 페이지

논리적 디바이스 정보

논리적 디바이스를 사용하면 애플리케이션 인스턴스 하나를 실행할 수 있습니다.

논리적 디바이스를 추가할 때는 애플리케이션 인스턴스 유형 및 버전 정의, 인터페이스 할당, 애플리케이션 구성으로 푸시되는 부트스트랩 설정 작업도 수행합니다.

독립형 논리적 디바이스와 클러스터된 논리적 디바이스

다음 논리적 디바이스 유형을 추가할 수 있습니다.

- 독립형 - 독립형 논리적 디바이스는 독립형 유닛이나 고가용성 쌍의 유닛으로 작동합니다.
- 클러스터 - 클러스터된 논리적 디바이스에서는 여러 유닛을 함께 그룹화할 수 있으므로 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. Firepower 9300에서는 3개 모듈 애플리케이션 인스턴스가 모두 단일 논리적 디바이스에 속합니다.



참고 Firepower 9300에서는 모든 모듈이 클러스터에 속해야 합니다. 한 보안 모듈에서 독립형 논리적 디바이스를 생성한 다음에 나머지 2개의 보안 모듈을 사용하는 클러스터를 생성할 수는 없습니다.

논리적 디바이스의 요구 사항 및 사전 요구 사항

요구 사항 및 사전 요구 사항은 다음 섹션을 참조하십시오.

클러스터링의 요구 사항 및 사전 요구 사항

새시 간 클러스터링 하드웨어 및 소프트웨어 요구 사항

클러스터의 모든 새시:

- 모든 보안 모듈이 동일한 유형이어야 합니다. 빈 슬롯을 포함하여 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 구성을 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. 인터페이스 모듈을 추가 또는 제거하거나 EtherChannel을 구성하는 등의 방법을 통해 클러스터링을 활성화한 후 FXOS에서 인터페이스를 변경하는 경우에는 각 새시에서 슬레이브 유닛부터 시작하여 마지막으로 마스터까지 같은 변경을 수행합니다.
- 동일한 NTP 서버를 사용해야 합니다. 시간을 수동으로 설정해서는 안 됩니다.
- ASA: 각 FXOS 새시를 License Authority 또는 Satellite Server에 등록해야 합니다. 슬레이브 유닛에 대한 추가 비용은 없습니다. Firepower Threat Defense의 경우 모든 라이선싱이 Firepower Management Center에서 처리됩니다.

새시 간 클러스터링을 위한 스위치 요구 사항

- Firepower 9300 새시에서 클러스터링을 구성하기 전에 스위치 구성을 완료하고 새시의 모든 EtherChannel을 스위치에 성공적으로 연결하십시오.
- 지원되는 스위치의 목록은 [Cisco FXOS 호환성](#)을 참고하십시오.

사이트 간 클러스터링을 위한 데이터 센터 인터커넥트 크기 조정

클러스터 제어 링크 트래픽을 처리하기 위한 DCI(data center interconnect) 대역폭을 다음 계산과 같이 예약해야 합니다.

$$\frac{\# \text{ 사이트당 클러스터 멤버의 수}}{2} \times \text{멤버당 클러스터 제어 링크 크기}$$

각 사이트의 멤버 수가 다를 경우, 더 큰 숫자를 계산에 사용합니다. DCI의 최소 대역폭은 한 멤버에 대한 클러스터 제어 링크의 크기보다 작으면 안 됩니다.

예를 들면 다음과 같습니다.

- 2개 사이트에 멤버가 4개인 경우:

- 총 클러스터 멤버 4개
- 각 사이트당 멤버 2개
- 멤버당 5Gbps 클러스터 제어 링크

$$\text{예약된 DCI 대역폭} = 5\text{Gbps}(2/2 \times 5\text{Gbps})$$

- 3개 사이트에 멤버가 6개인 경우 크기가 다음과 같이 증가함:

- 총 클러스터 멤버 6개
- 사이트 1에 멤버 3개, 사이트 2에 멤버 2개, 사이트 3에 멤버 1개
- 멤버당 10Gbps 클러스터 제어 링크

$$\text{예약된 DCI 대역폭} = 15\text{Gbps}(3/2 \times 10\text{Gbps})$$

- 2개 사이트에 멤버가 2개인 경우:

- 총 클러스터 멤버 2개
- 사이트당 멤버 1개
- 멤버당 10Gbps 클러스터 제어 링크

예약된 DCI 대역폭 = 10Gbps(1/2 x 10Gbps = 5Gbps). 그러나 최소 대역폭은 클러스터 제어 링크의 크기(10Gbps)보다 작으면 안 됩니다.

논리적 디바이스 관련 지침 및 제한 사항

지침 및 제한 사항은 다음 섹션을 참조하십시오.

일반 지침 및 제한 사항

방화벽 모드

FTD에 대해 부트스트랩 구성에서 방화벽 모드를 라우팅 또는 투명으로 설정할 수 있습니다. ASA의 경우 구축 후에 방화벽 모드를 투명으로 변경할 수 있습니다. [ASA를 투명 방화벽 모드로 변경, 141 페이지](#) 섹션을 참조하십시오.

고가용성

- 애플리케이션 구성 내에서 고가용성을 구성합니다.
- 모든 데이터 인터페이스를 장애 조치 및 상태 링크로 사용할 수 있습니다.
- 자세한 내용은 고가용성에 대한 애플리케이션 구성 가이드 장을 참조하십시오.

상황 모드

- 다중 상황 모드는 ASA에서만 지원됩니다.
- 구축 후에 ASA에서 다중 상황 모드를 활성화합니다.

클러스터링 지침 및 제한 사항

새시 간 클러스터링을 위한 스위치

- ASR 9006의 경우 기본이 아닌 MTU를 설정하려면 ASR 인터페이스 MTU를 클러스터 디바이스 MTU보다 14바이트 높게 설정합니다. 그렇지 않으면, **mtu-ignore** 옵션을 사용하지 않는 경우 OSPF 인접 피어링 시도에 실패할 수 있습니다. 클러스터 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. Nexus Series와 같은 일부 스위치는 ISSU(서비스 내 소프트웨어 업그레이드) 수행 시 고속 LACP가 지원되지 않으므로 클러스터링에서는 ISSU를 사용하지 않는 것이 좋습니다.
- 스위치에서는 **source-dest-ip** 또는 **source-dest-ip-port** EtherChannel 로드 밸런싱 알고리즘 중 하나를 사용하는 것이 좋습니다(Cisco Nexus OS 및 Cisco IOS **port-channel load-balance** 명령 참조). 로드 밸런싱 알고리즘에서는 **vlan** 키워드를 사용하지 마십시오. 이렇게 할 경우 클러스터의 디바이스에 트래픽이 균일하지 않게 분산될 수 있습니다.
- 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.

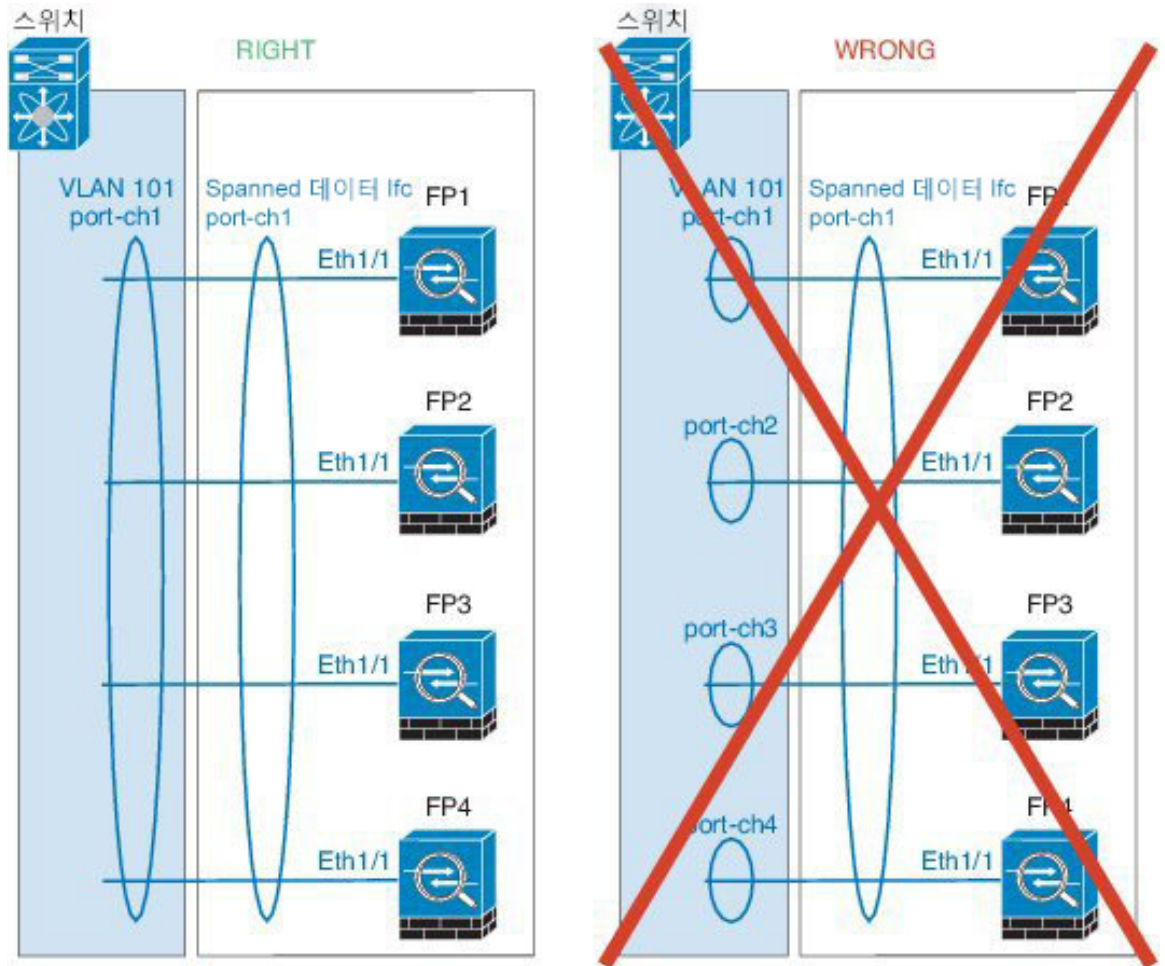
- 클러스터 제어 링크 경로의 스위치에서는 L4 체크섬을 확인하지 않습니다. 클러스터 제어 링크를 통해 리디렉션된 트래픽에는 올바른 L4 체크섬이 없습니다. L4 체크섬을 확인하는 스위치의 경우 트래픽이 감소하는 결과를 초래할 수 있습니다.
- 포트 채널 번들링 다운타임은 구성된 `keepalive` 기간을 초과하면 안 됩니다.
- Supervisor 2T EtherChannel에서 기본 해시 분산 알고리즘은 적응형입니다. VSS 설계에서 비대칭 트래픽을 방지하려면 클러스터 디바이스에 연결된 포트 채널의 해시 알고리즘을 다음과 같이 변경하여 수정합니다.

```
router(config) # port-channel id hash-distribution fixed
```

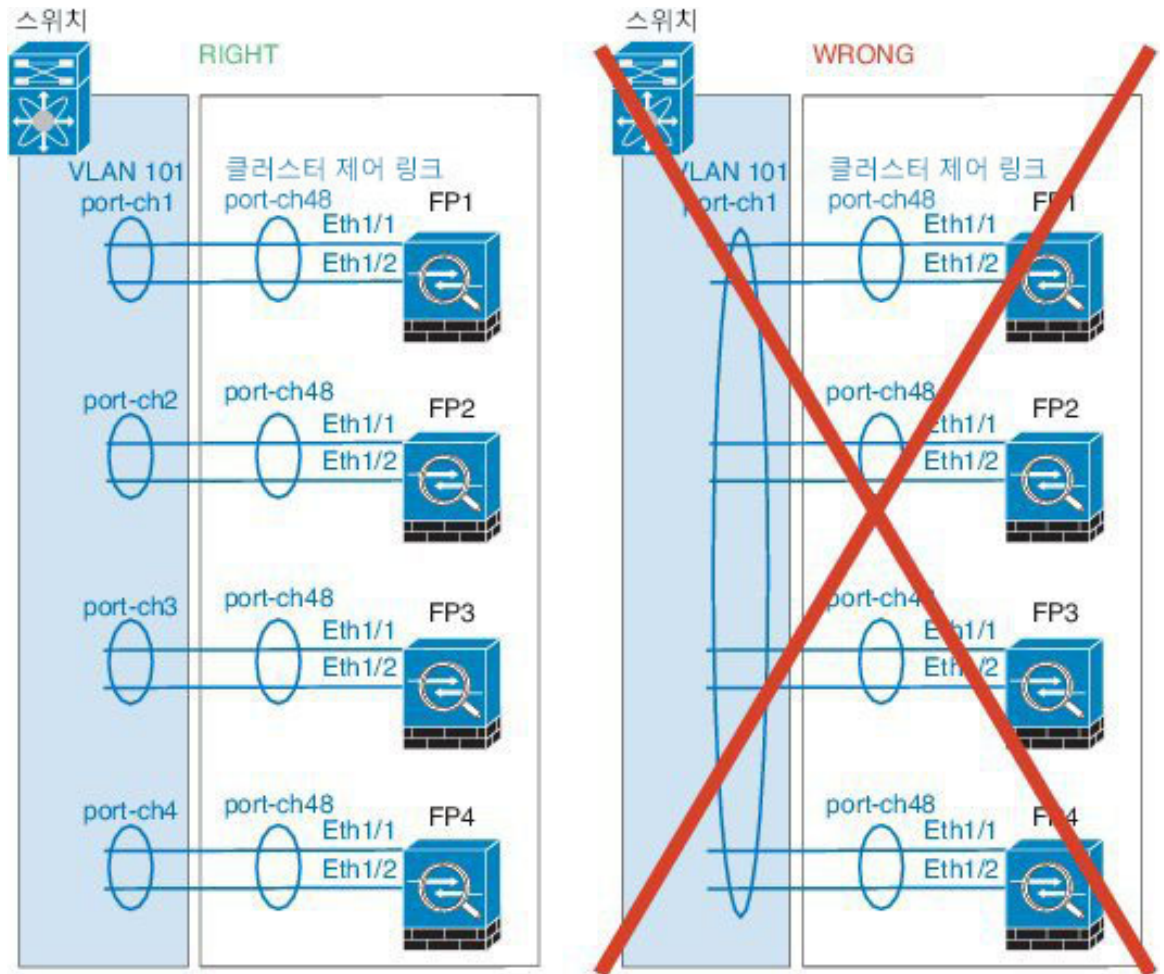
VSS 피어 링크의 적응형 알고리즘을 활용할 때가 있을 수 있으므로 알고리즘을 전역으로 변경하지 마십시오.

새시 간 클러스터링을 위한 EtherChannel

- 연결 스위치의 경우, EtherChannel 모드를 활성으로 설정합니다. On(켜기) 모드는 Firepower 9300 새시에서 지원되지 않으며 클러스터 제어 링크에서도 지원되지 않습니다.
- FXOS EtherChannel에서는 기본적으로 LACP 속도가 `fast`(고속)로 설정됩니다. Nexus Series와 같은 일부 스위치는 ISSU(서비스 내 소프트웨어 업그레이드) 수행 시 고속 LACP가 지원되지 않으므로 클러스터링에서는 ISSU를 사용하지 않는 것이 좋습니다.
- 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 구성과 디바이스-로컬 EtherChannel 구성 — Spanned EtherChannel과 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.



사이트 간 클러스터링

사이트 간 클러스터링에 대한 다음 지침을 참조하십시오.

- 클러스터 제어 링크 레이턴시는 RTT(왕복 시간)가 20ms 이하여야 합니다.
- 클러스터 제어 링크는 오류가 나거나 폐기된 패킷이 없는 안정적인 상태여야 합니다. 예를 들어, 전용 링크를 사용해야 합니다.
- 연결 리밸런싱을 구성하지 마십시오. 이렇게 할 경우 다른 사이트의 클러스터 멤버에 연결이 리밸런싱됩니다.
- 클러스터를 구현할 경우 들어오는 연결에 대한 여러 사이트에 있는 멤버가 구분되지 않습니다. 따라서 하나의 특정한 연결의 연결 역할은 사이트 전체를 포괄하게 될 수 있습니다. 이는 정상적인 동작입니다.
- 투명 모드에서, 클러스터가 내부 및 외부 라우터(north-south 삽입이라고도 함) 쌍 사이에 위치하면 내부 라우터 모두에서 MAC 주소를 공유해야 하며 외부 라우터 모두에서도 MAC 주소를 공유해야 합니다. 사이트 1의 클러스터 멤버가 사이트 2의 멤버에 연결을 전달할 경우, 목적지 MAC

주소가 유지됩니다. MAC 주소가 사이트 1의 라우터와 동일할 경우 패킷은 사이트 2의 라우터에만 도달합니다.

- 투명 모드에서 클러스터가 내부 네트워크(East-West 삽입이라고 함) 사이에서 방화벽을 위해 각 사이트에서 데이터 네트워크 및 게이트웨이 라우터 사이에 위치하면 각 게이트웨이 라우터는 HSRP와 같은 첫 번째 홉 이중화 프로토콜(FHRP)을 사용하여 각 사이트에서 동일한 가상 IP 및 MAC 주소 대상을 제공해야 합니다. 데이터 VLAN은 OTV(오버레이 전송 가상화) 또는 유사한 기능을 사용하는 사이트 전체로 확장됩니다. DCI를 통해 다른 사이트로 전송 중인 로컬 게이트웨이 라우터에 예약된 트래픽을 방지하려면 필터를 생성해야 합니다. 게이트웨이 라우터가 1개의 사이트에 연결할 수 없게 되면, 모든 필터를 제거해야 트래픽이 성공적으로 다른 사이트의 게이트웨이에 연결할 수 있습니다.
- Spanned EtherChannel을 사용하는 라우팅 모드인 경우 사이트별 MAC 주소를 구성하십시오. OTV 또는 유사한 것을 사용하여 사이트 전체로 데이터 VLAN을 확장하십시오. 전역 MAC 주소로 향하는 트래픽이 DCI를 통해 다른 사이트에 가지 않도록 필터를 생성해야 합니다. 어떤 사이트에서 클러스터가 연결할 수 없게 되면 트래픽이 다른 사이트의 클러스터 유닛에 성공적으로 도달할 수 있도록 모든 필터를 제거해야 합니다. 사이트 간 클러스터가 확장 세그먼트의 FHR(First Hop Router)로 작동하는 경우에는 동적 라우팅이 지원되지 않습니다.

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.

기본값

클러스터 제어 링크는 Port-channel 48을 사용합니다.

독립형 논리적 디바이스 추가

독립형 논리적 디바이스는 단독으로 사용하거나 고가용성 유닛으로 사용할 수 있습니다. 고가용성 사용 방식에 대한 자세한 내용은 [고가용성 쌍 추가, 125 페이지](#) 섹션을 참조하십시오.

독립형 ASA 추가

독립형 논리적 디바이스는 단독으로 작동하거나 고가용성 쌍에서 작동합니다. Firepower 9300과 같이 모듈이 여러 개인 디바이스에서는 클러스터 또는 독립형 디바이스를 구축할 수 있습니다. 클러스터는 모든 모듈을 사용해야 하므로 모듈이 2개인 클러스터와 단일 독립형 디바이스를 혼합하는 등의 방식은 상용할 수 없습니다.

Firepower 9300 새시에서 라우팅된 방화벽 모드 방화벽 모드 ASA를 구축할 수 있습니다. ASA를 투명 방화벽 모드로 변경하려면 이 절차를 완료한 후에 [ASA를 투명 방화벽 모드로 변경, 141 페이지](#) 섹션을 참조하십시오.

다중 컨텍스트 모드인 경우 먼저 논리적 디바이스를 구축한 다음 ASA 애플리케이션에서 다중 컨텍스트 모드를 활성화해야 합니다.

시작하기 전에

- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드(Cisco.com에서 이 이미지 다운로드, 46 페이지 참조)한 다음 해당 이미지를 Firepower 9300 새시에 다운로드합니다 (논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 9300 새시, 49 페이지 참조).
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 관리 인터페이스는 필수 항목입니다. 이 관리 인터페이스는 새시 관리용으로만 사용되는 새시 관리 인터페이스(FXOS에서 MGMT, management0 또는 기타 유사한 이름으로 표시될 수 있으며)와는 다릅니다.

프로시저

단계 1 Security Services(보안 서비스) 모드를 설정합니다.

scope ssa

예제:

```
Firepower# scope ssa
Firepower /ssa #
```

단계 2 애플리케이션 인스턴스 이미지 버전을 설정합니다.

a) 사용 가능한 이미지를 확인합니다. 사용하려는 버전 번호를 적어 둡니다.

show app

예제:

```
Firepower /ssa # show app
  Name          Version      Author      Supported Deploy Types CSP Type      Is Default
  App
  -----
  asa           9.9.1       cisco      Native      Application No
  asa           9.10.1      cisco      Native      Application Yes
  ftd           6.2.3       cisco      Native      Application Yes
```

b) 보안 모듈/엔진 슬롯에 범위를 설정합니다.

scope slot slot_id

slot_id는 Firepower 9300의 경우 1, 2 또는 3입니다.

예제:

```
Firepower /ssa # scope slot 1
Firepower /ssa/slot #
```

c) 애플리케이션 인스턴스를 생성합니다.

enter app-instance asa

예제:

```
Firepower /ssa/slot # enter app-instance asa
Firepower /ssa/slot/app-instance* #
```

- d) ASA 이미지 버전을 설정합니다.

set startup-version version

예제:

```
Firepower /ssa/slot/app-instance* # set startup-version 9.10.1
```

- e) 슬롯 모드를 종료합니다.

exit

예제:

```
Firepower /ssa/slot/app-instance* # exit
Firepower /ssa/slot* #
```

- f) ssa 모드를 종료합니다.

exit

예제:

```
Firepower /ssa/slot* # exit
Firepower /ssa* #
```

예제:

```
Firepower /ssa # scope slot 1
Firepower /ssa/slot # enter app-instance asa
Firepower /ssa/slot/app-instance* # set startup-version 9.10.1
Firepower /ssa/slot/app-instance* # exit
Firepower /ssa/slot* # exit
Firepower /ssa* #
```

- 단계 3 논리적 디바이스를 생성합니다.

enter logical-device device_name asa slot_id standalone

예제:

```
Firepower /ssa # enter logical-device ASA1 asa 1 standalone
Firepower /ssa/logical-device* #
```

- 단계 4 논리적 디바이스에 관리 및 데이터 인터페이스를 할당합니다. 각 인터페이스에 대해 이 작업을 반복합니다.

create external-port-link *name interface_id asa*

set description *description*

exit

- *name*(이름) - ASA 구성에서 사용되는 인터페이스 이름이 아닌 Firepower 9300 새시 Supervisor(관리자)가 사용하는 이름입니다.
- *description*(설명) - 공백이 있는 구는 따옴표(")로 묶습니다.

예제:

```
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
```

단계 5 관리 부트스트랩 정보를 구성합니다.

a) 부트스트랩 개체를 생성합니다.

create mgmt-bootstrap asa

예제:

```
Firepower /ssa/logical-device* # create mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

b) 관리자 활성화를 지정합니다.

create bootstrap-key-secret PASSWORD

set value

password 값을 입력합니다.

password 값을 확인합니다.

exit

예제:

비밀번호를 복구할 때는 사전 구성된 ASA 관리자를 사용하면 유용합니다. FXOS 액세스 권한이 있다면 관리자 비밀번호를 잊어버린 경우 재설정할 수 있습니다.

예제:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create bootstrap-key-secret PASSWORD
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Enter a value: floppylampshade
Confirm the value: floppylampshade
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
```

```
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

- c) IPv4 관리 인터페이스 설정을 구성합니다.

```
create ipv4 slot_id default  
set ip ip_address mask network_mask  
set gateway gateway_address  
exit
```

예제:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv4 1 default  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set ip 10.10.10.34 mask 255.255.255.0  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # set gateway 10.10.10.1  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv4* # exit  
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

- d) IPv6 관리 인터페이스 설정을 구성합니다.

```
create ipv6 slot_id default  
set ip ip_address prefix-length prefix  
set gateway gateway_address  
exit
```

예제:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # create ipv6 1 default  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # set ip 2001:0DB8:BA98::3210  
prefix-length 64  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # set gateway 2001:0DB8:BA98::3211  
Firepower /ssa/logical-device/mgmt-bootstrap/ipv6* # exit  
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

- e) 관리 부트스트랩 모드를 종료합니다.

```
exit
```

예제:

```
Firepower /ssa/logical-device/mgmt-bootstrap* # exit  
Firepower /ssa/logical-device* #
```

단계 6 구성을 저장합니다.

```
commit-buffer
```

예제:

```
Firepower /ssa/logical-device* # commit-buffer
```

```
Firepower /ssa/logical-device #
```

예

```
Firepower# scope ssa
Firepower /ssa # scope slot 1
Firepower /ssa/slot # enter app-instance asa
Firepower /ssa/slot/app-instance* # set startup-version 9.10.1
Firepower /ssa/slot/app-instance* # exit
Firepower /ssa/slot* # exit
Firepower /ssa* # create logical-device MyDevice1 asa 1 standalone
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
Enter a value: secretglassine
Confirm the value: secretglassine
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 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/bootstrap-key* # commit-buffer
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key #
```

고가용성 쌍 추가

ASA 고가용성(장애 조치)은 FXOS가 아닌 애플리케이션 내에서 구성됩니다. 그러나 고가용성을 사용할 수 있도록 새시를 준비하려는 경우 다음 단계를 참조하십시오.

시작하기 전에

- 고가용성 시스템 요구 사항은 고가용성을 위한 애플리케이션 구성 가이드 장을 참조하십시오.

프로시저

단계 1 각 논리적 디바이스는 별도의 새시에 있어야 합니다. Firepower 9300의 경우 새시 내 고가용성은 지원되지 않을 수 있으며 사용하지 않는 것이 좋습니다.

단계 2 각 논리적 디바이스에 동일한 인터페이스를 할당합니다.

단계 3 장애 조치 및 상태 링크용으로 데이터 인터페이스 1~2개를 할당합니다.

이러한 인터페이스는 두 새시 간의 고가용성 트래픽을 교환합니다. 장애 조치 및 상태 링크를 함께 사용하려면 10GB 데이터 인터페이스를 사용하는 것이 좋습니다. 사용 가능한 인터페이스가 있다면 장애 조치와 상태 링크를 각기 별도로 사용할 수 있습니다. 상태 링크에는 최대 대역폭이 필요합니다. 관리 유형 인터페이스는 장애 조치 또는 상태 링크용으로 사용할 수 없습니다. 장애 조치 인터페이스와 같은 네트워크 세그먼트에 다른 디바이스가 없는 상태로 새시 간에 스위치를 사용하는 것이 좋습니다.

단계 4 논리적 디바이스에서 고가용성을 활성화합니다.

단계 5 고가용성을 활성화한 후에 인터페이스를 변경해야 하는 경우에는 먼저 스탠바이 유닛에서 변경을 수행한 다음 액티브 유닛에서 변경을 수행합니다.

참고 ASA의 경우 네트워크 모듈/EtherChannel을 제거하거나 EtherChannel에 인터페이스를 재할당하는 등 FXOS에서 인터페이스를 제거하면 ASA 구성에서 원래 명령이 유지되므로 필요한 조정을 수행할 수 있습니다. 구성에서 인터페이스를 제거하는 경우에는 구성 전반에 걸쳐 영향을 줄 수 있습니다. ASA OS에서 이전 인터페이스 구성을 수동으로 제거할 수 있습니다.

클러스터 추가

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.



참고 FTD은 여러 새시 전반에(새시 간) 클러스터를 지원하지 않으며 새시 내 클러스터링만 지원됩니다.

클러스터링 정보 Firepower 9300 새시

클러스터는 단일 논리적 유닛으로 작동하는 여러 개의 디바이스로 구성됩니다. Firepower 9300 새시에서 클러스터를 구축할 때는 다음 작업이 수행됩니다.

- 유닛 간 통신에 사용되는 클러스터 제어 링크(기본값: port-channel 48)를 생성합니다. 인트라 새시 클러스터링(intra-chassis clustering)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 구성을 생성합니다.

클러스터를 구축할 때, Firepower 9300 새시 Supervisor(관리자)는 클러스터 이름, 클러스터 제어 링크 인터페이스 및 기타 클러스터 설정을 포함하는 각 유닛에 최소한의 부트스트랩 구성을 푸시합니다. 클러스터링 환경을 사용자 정의하려는 경우, 사용자가 일부 부트스트랩 구성을 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. `conrefFirepower 9300 Supervisor`(관리자)는 EtherChannel 기술을 내부에 사용하여 트래픽을 공유 인터페이스의 다중 모듈에 로드 밸런싱하므로 모든 데이터 인터페이스 유형이 *Spanned* 모드에서 작동합니다. 새시 간 클러스터링의 경우, 모든 데이터 인터페이스에 *Spanned EtherChannel*을 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 구성을 수행해야 하며 이후에 구성은 보조 유닛에 복제됩니다.

Cluster Control Link

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 경우에는 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.
- 멤버가 변경된 경우, 클러스터에서는 다량의 연결을 다시 밸런싱해야 하므로 일시적으로 많은 양의 클러스터 제어 링크 대역폭을 사용합니다.

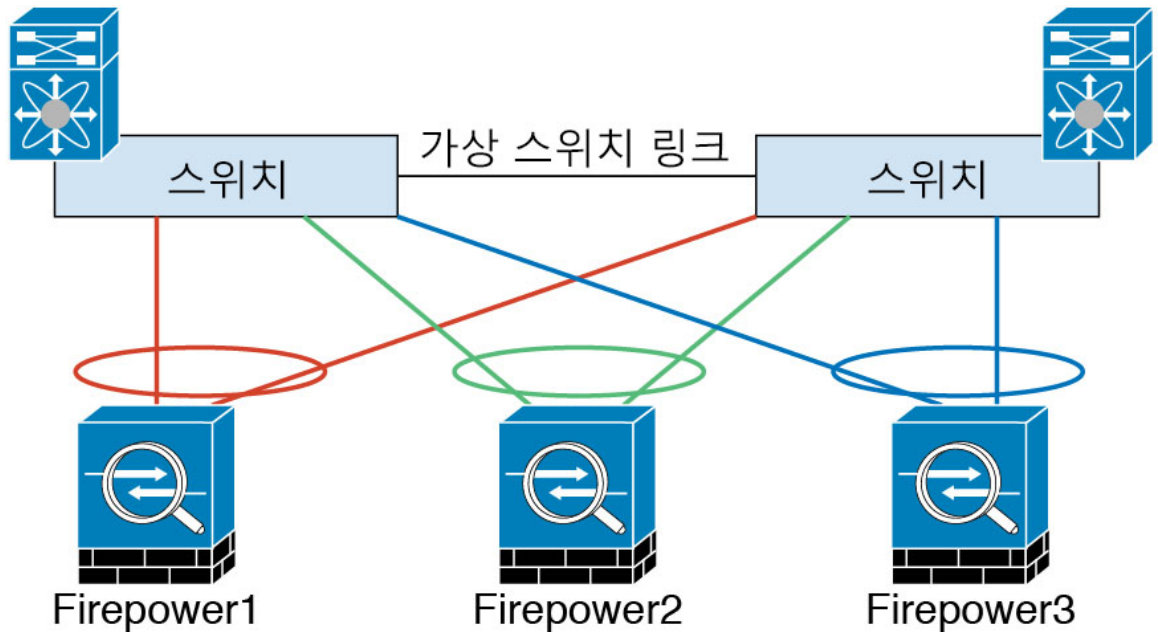
대역폭이 높은 클러스터 제어 링크를 사용하면 멤버가 변경될 경우 클러스터를 더 빠르게 통합할 수 있고 처리량 병목 현상을 방지할 수 있습니다.



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 Firepower 9300 인터페이스를 동일한 EtherChannel 내에서 연결하여 VSS 또는 vPC의 스위치와 별도로 분리할 수 있습니다. 이러한 별도의 스위치는 단일 스위치 역할을 하므로, 스위치 인터페이스는 동일한 EtherChannel 포트 채널 인터페이스의 멤버입니다. 이러한 EtherChannel은 디바이스 로컬이 아닌 스펠 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.

클러스터 제어 링크 네트워크

Firepower 9300 새시에서는 새시 ID 및 슬롯 ID `127.2.chassis_id.slot_id`를 기준으로 하여 각 유닛에 대해 클러스터 제어 링크 인터페이스 IP 주소를 자동 생성합니다. FXOS 또는 애플리케이션 내에서 이 IP 주소를 수동으로 설정할 수는 없습니다. 클러스터 제어 링크 네트워크는 유닛 간에 라우터를 포함할 수 없으며 레이어 2 스위칭만 허용됩니다. 사이트 간 트래픽의 경우에는 OTV(Overlay Transport Virtualization)를 사용하는 것이 좋습니다.

관리 네트워크

모든 유닛을 단일한 관리 네트워크에 연결할 것을 권장합니다. 이 네트워크는 클러스터 제어 링크와 분리되어 있습니다.

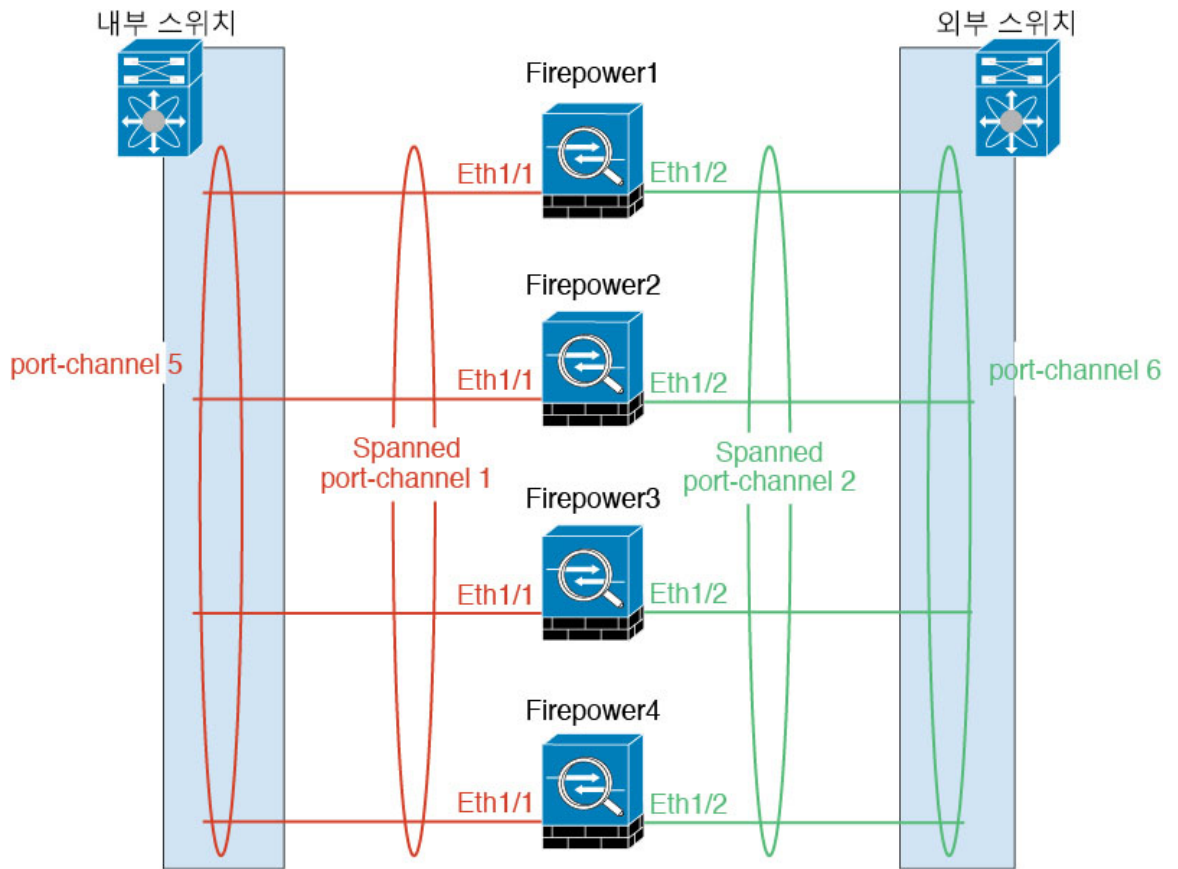
관리 인터페이스

클러스터에 관리 유형 인터페이스를 할당해야 합니다. 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. 또한 주소의 범위를 구성하여 현재 기본 유닛을 비롯한 각 유닛에서 해당 범위의 로컬 주소를 사용할 수 있도록 해야 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 트러블슈팅에도 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.

Spanned EtherChannels

새시당 하나 이상의 인터페이스를 클러스터 내의 모든 새시를 포괄하는 EtherChannel로 그룹화할 수 있습니다. EtherChannel에서는 채널에서 사용 가능한 모든 활성 인터페이스 전반의 트래픽을 취합합니다. 스패 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 간 클러스터링

사이트 간 설치의 경우 권장 지침을 준수하여 클러스터링을 활용할 수 있습니다.

각 클러스터 새시를 별도의 사이트 ID에 속하도록 구성할 수 있습니다.

사이트 ID는 사이트별 MAC 주소. 클러스터에서 온 패킷은 사이트별 MAC 주소, 클러스터가 수신한 패킷은 전역 MAC 주소. 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 Spanned EtherChannel만을 사용하는 라우팅 모드에서 지원 됩니다.

사이트 ID는 LISP 검사를 사용한 플로우 모빌리티 활성화.

사이트 간 클러스터링에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- 데이터 센터 인터커넥트 크기 조정 -클러스터링의 요구 사항 및 사전 요구 사항, 114 페이지
- 사이트 간 지침 -클러스터링 지침 및 제한 사항, 116 페이지
- 사이트 간 예시 -사이트 간 클러스터링 예시, 146 페이지

ASA 클러스터에 추가

단일 Firepower 9300 새시를 새시 간 클러스터로 추가하거나 새시 간 클러스터링용으로 여러 새시를 추가할 수 있습니다. 새시 간 클러스터링의 경우 각 새시를 개별적으로 구성해야 합니다. 새시 하나에 클러스터를 추가한 다음 대부분의 동일 설정을 다음 새시에 입력합니다.

ASA 클러스터 생성

Firepower 9300 새시에서 클러스터를 구축합니다.

다중 컨텍스트 모드의 경우 먼저 논리적 디바이스를 구축한 다음 ASA 애플리케이션에서 다중 컨텍스트 모드를 활성화해야 합니다.

Firepower 9300 새시에서 라우팅된 방화벽 모드 방화벽 모드 ASA를 구축할 수 있습니다. ASA를 투명 방화벽 모드로 변경하려면 초기 구축을 완료한 다음 ASA CLI 내에서 방화벽 모드를 변경합니다.

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.

프로시저

-
- 단계 1** 클러스터를 구축하기 전에 데이터 유형 인터페이스 또는 EtherChannel(포트 채널)을 최소 1개 구성합니다. [EtherChannel\(포트 채널\) 추가, 107 페이지](#) 또는 [실제 인터페이스 구성, 105 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. [EtherChannel\(포트 채널\) 추가, 107 페이지](#) 또는 [실제 인터페이스 구성, 105 페이지](#)를 참조하십시오.
- 관리 인터페이스는 필수 항목입니다. 이 관리 인터페이스는 새시 관리용으로만 사용되는 새시 관리 인터페이스와는 다릅니다. FXOS에서는 새시 관리 인터페이스가 MGMT, management0 또는 기타 유사한 이름으로 표시될 수 있습니다.
- 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 slots clustered

- *device_name* - Firepower 9300 새시 Supervisor(관리자)가 클러스터링 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 보안 모듈 구성에 사용되는 클러스터 이름이 아닙니다. 하드웨어를 아직 설치하지 않은 경우에도 보안 모듈 3개를 모두 지정해야 합니다.
- *slots* - 새시 모듈을 클러스터에 할당합니다. Firepower 4100의 경우 1을 지정합니다. Firepower 9300의 경우 1,2,3을 지정합니다. 모듈을 설치하지 않은 경우에도 Firepower 9300 새시의 3개 모듈 슬롯 모두에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

예제:

```
Firepower /ssa # enter logical-device ASA1 asa 1,2,3 clustered
Firepower /ssa/logical-device* #
```

단계 6 관리 부트스트랩 개체를 생성합니다.

enter mgmt-bootstrap asa

예제:

```
Firepower /ssa/logical-device* # enter mgmt-bootstrap asa
Firepower /ssa/logical-device/mgmt-bootstrap* #
```

단계 7 관리자 비밀번호를 지정합니다.

enter bootstrap-key-secret PASSWORD**set value****exit****exit**

비밀번호를 복구할 때는 사전 구성된 ASA 관리자가 있으면 유용합니다. FXOS 액세스 권한이 있다면 관리자 비밀번호를 잊어버린 경우 재설정할 수 있습니다.

예제:

```
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # set value
Enter a value: happytuesday
Confirm the value: happytuesday
Firepower /ssa/logical-device/mgmt-bootstrap/bootstrap-key-secret* # exit
Firepower /ssa/logical-device/mgmt-bootstrap* # exit
```

```
Firepower /ssa/logical-device* #
```

단계 8 클러스터 매개변수를 구성합니다.

enter cluster-bootstrap

예제:

```
Firepower /ssa/logical-device* # enter cluster-bootstrap
Firepower /ssa/logical-device/cluster-bootstrap* #
```

단계 9 보안 모듈 구성에서 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 10 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 11 관리 IP 주소 정보를 구성합니다.

이 정보는 보안 모듈 구성의 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 마스터 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 12 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 13 클러스터 제어 링크의 제어 트래픽에 대해 인증 키를 구성합니다.

```
set key
```

예제:

```
Firepower /ssa/logical-device/cluster-bootstrap* # set key
Key: diamonddogs
```

공유 비밀을 입력하라는 프롬프트가 표시됩니다.

공유 비밀은 1자 ~ 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 14 클러스터 부트스트랩 모드 및 논리적 디바이스 모드를 종료합니다.

```
exit
```

```
exit
```

단계 15 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 16 구성을 커밋합니다.

```
commit-buffer
```

Firepower 9300 새시 Supervisor(관리자)는 기본 보안 모듈 소프트웨어 버전을 다운로드하고 클러스터 부트스트랩 구성 및 관리 인터페이스 설정을 각 보안 모듈에 입력하여 클러스터를 구축합니다.

단계 17 클러스터에 다른 새시를 추가하려면 고유한 **chassis-id**를 구성해야 하는 경우를 제외하고 이 절차를 반복합니다. 아니면 두 새시 모두에 동일한 구성을 사용합니다.

인터페이스 구성은 새 새시에서 동일해야 합니다. FXOS 새시 구성 내보내기과 가져오기를 통해 이 프로세스를 더 쉽게 수행할 수 있습니다.

단계 18 마스터 유닛 ASA에 연결하여 클러스터링 구성을 맞춤 설정합니다.

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

클러스터 멤버 더 추가

ASA 클러스터 멤버를 추가하거나 교체합니다.



참고 이 절차는 새시 추가 또는 교체 시에만 적용됩니다. 클러스터링이 이미 활성화된 Firepower 9300에 모듈을 추가하거나 교체하는 경우에는 모듈이 자동으로 추가됩니다.

시작하기 전에

- 기존 클러스터의 관리 IP 주소 풀에 이 새 멤버용 IP 주소가 충분히 포함되어 있는지 확인하십시오. IP 주소가 충분하지 않으면 이 새 멤버를 추가하기 전에 각 새시에서 기존 클러스터 부트스트랩 구성을 수정해야 합니다. 이 변경을 수행하는 경우 논리적 디바이스를 재시작해야 합니다.
- 인터페이스 구성은 새 새시에서 동일해야 합니다. FXOS 새시 구성 내보내기과 가져오기를 통해 이 프로세스를 더 쉽게 수행할 수 있습니다.
- 다중 컨텍스트 모드의 경우 첫 번째 클러스터 멤버의 ASA 애플리케이션에서 다중 컨텍스트 모드를 활성화합니다. 그러면 추가 클러스터 멤버가 다중 컨텍스트 모드 구성을 자동으로 상속합니다.

프로시저

클러스터에 다른 새시를 추가하려면 고유한 **chassis-id**를 구성해야 하는 경우를 제외하고 [ASA 클러스터 생성, 131 페이지](#)의 절차를 반복합니다. 아니면 새 새시에 동일한 구성을 사용합니다.

논리적 디바이스 관리

논리적 디바이스를 삭제하고, ASA를 투명 모드로 변환하고, 인터페이스 구성을 변경하고, 기존 논리적 디바이스에서 기타 작업을 수행할 수 있습니다.

애플리케이션 콘솔에 연결

다음 절차를 수행하여 애플리케이션의 콘솔에 연결합니다.

프로시저

단계 1 모듈 CLI에 연결합니다.

connect module slot_number console

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

예제:

```
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>
```

단계 **2** 애플리케이션 콘솔에 연결합니다.

connect asa

예제:

```
Firepower-module1> connect asa
Connecting to asa(asal) console... hit Ctrl + A + D to return to bootCLI
[...]
asa>
```

예제:

```
Firepower-module1> connect ftd
Connecting to ftd(ftd-native) console... enter exit to return to bootCLI
[...]
>
```

단계 **3** 애플리케이션 콘솔을 FXOS 모듈 CLI로 종료합니다.

- ASA - **Ctrl-a, d**를 입력합니다.

문제 해결을 위해 FXOS 모듈 CLI를 사용할 수 있습니다.

단계 **4** FXOS CLI의 Supervisor(관리자) 수준으로 돌아갑니다.

a) ~를 입력합니다.

텔넷 애플리케이션을 종료합니다.

b) 텔넷 애플리케이션을 종료하려면 다음을 입력합니다.

```
telnet>quit
```

예시

다음 예시에서는 보안 모듈 1에 있는 ASA에 연결한 다음 FXOS CLI의 Supervisor(관리자) 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

프로시저

단계 1 보안 서비스 모드를 입력합니다.

```
Firepower# scope ssa
```

단계 2 새시에 있는 논리적 디바이스에 대한 세부사항을 확인합니다.

```
Firepower /ssa # show logical-device
```

단계 3 삭제할 논리적 디바이스 각각에 대해 다음 명령을 입력합니다.

```
Firepower /ssa # delete logical-device device_name
```

단계 4 논리적 디바이스에 설치되어 있는 애플리케이션에 대한 세부사항을 확인합니다.

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

단계 5 삭제할 애플리케이션 각각에 대해 다음 명령을 입력합니다.

- a) Firepower /ssa # **scope slot slot_number**
- b) Firepower /ssa/slot # **delete app-instance application_name**
- c) Firepower /ssa/slot # **exit**

단계 6 구성을 커밋합니다.

```
commit-buffer
```

시스템 구성에 트랜잭션을 커밋합니다.

예시

```

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

```

ASA를 투명 방화벽 모드로 변경

라우팅된 방화벽 모드 ASA는 Firepower 9300 새시에서만 구축할 수 있습니다. ASA를 투명 방화벽 모드로 변경하려면 초기 구축을 완료한 다음 ASA CLI 내에서 방화벽 모드를 변경합니다. 독립형 ASA의 경우 방화벽 모드를 변경하면 구성이 지워지므로 Firepower 9300 새시에서 구성을 재구축하여 부트스트랩 구성을 다시 가져와야 합니다. 그러면 ASA는 투명 모드로 유지되며 부트스트랩 구성도 계속 작동합니다. 클러스터된 ASA의 경우에는 구성이 지워지지 않으므로 FXOS에서 부트스트랩 구성을 재구축하지 않아도 됩니다.

프로시저

- 단계 1 애플리케이션 콘솔에 연결, 138 페이지에 따라 ASA 콘솔에 연결합니다. 클러스터의 경우 기본 유닛에 연결합니다. 장애 조치 쌍의 경우 액티브 유닛에 연결합니다.
- 단계 2 구성 모드를 설정합니다.

enable

configure terminal

기본적으로 enable 비밀번호는 비어 있습니다.

단계 3 방화벽을 투명 모드로 설정합니다.

firewall transparent

단계 4 구성을 저장합니다.

write memory

클러스터 또는 장애 조치 쌍의 경우 이 구성이 보조 유닛에 복제됩니다.

```
asa(config)# firewall transparent
asa(config)# write memory
Building configuration...
Cryptochecksum: 9f831dfb 60dffa8c 1d939884 74735b69

3791 bytes copied in 0.160 secs
[OK]
asa(config)#
Beginning configuration replication to Slave unit-1-2
End Configuration Replication to slave.

asa(config)#
```

단계 5 Firepower Chassis Manager **Logical Devices**(논리적 디바이스) 페이지에서 **Edit**(수정) 아이콘을 클릭하여 ASA를 수정합니다.

Provisioning(프로비저닝) 페이지가 나타납니다.

단계 6 디바이스 아이콘을 클릭하여 부트스트랩 구성을 수정합니다. 구성의 값을 변경한 후에 **OK**(확인)를 클릭합니다.

Password(비밀번호) 필드 등 하나 이상의 필드 값을 변경해야 합니다.

부트스트랩 구성 변경에 대한 경고가 표시되면 **Yes**(예)를 클릭합니다.

단계 7 **Save**(저장)를 클릭하여 ASA에 구성을 재구축합니다. 새시 간 클러스터 또는 장애 조치 쌍의 경우 5~7 단계를 반복하여 각 새시에서 부트스트랩 구성을 재구축합니다.

새시/보안 모듈이 다시 로드되고 ASA가 다시 작동할 때까지 몇 분 정도 기다립니다. 이제 ASA는 작동하는 부트스트랩 구성을 포함하지만 투명 모드로 유지됩니다.

Firepower Threat Defense 논리적 디바이스의 인터페이스 변경

Firepower Threat Defense 논리적 디바이스에서 관리 인터페이스를 할당, 할당 해제할 수 있습니다. 그런 다음 Firepower Management Center에서 인터페이스 구성을 동기화할 수 있습니다.

시작하기 전에

- **실제 인터페이스 구성, 105 페이지** 및 **EtherChannel**(포트 채널) 추가, **107 페이지**에 따라 인터페이스를 구성하고 EtherChannel을 추가합니다.

- 논리적 디바이스에 영향을 주거나 Firepower Management Center에서 동기화를 수행하지 않고도 할당된 EtherChannel의 멤버십을 수정할 수 있습니다.
- 모든 인터페이스가 기본적으로 클러스터에 할당된 경우와 같이 이미 할당된 인터페이스를 EtherChannel에 추가하려는 경우에는 먼저 논리적 디바이스에서 인터페이스 할당을 해제한 다음 EtherChannel에 인터페이스를 추가해야 합니다. 새 EtherChannel의 경우 이렇게 한 후에 디바이스에 EtherChannel을 할당할 수 있습니다.
- 관리 또는 Firepower 이벤트 인터페이스를 교체하려는 경우 Firepower Chassis Manager를 사용해야 합니다. CLI에서는 이 변경을 지원하지 않습니다.
- 클러스터링 또는 고가용성의 경우에는 Firepower Management Center에서 구성을 동기화하기 전에 모든 유닛에서 인터페이스를 추가하거나 제거해야 합니다. 먼저 슬레이브/스탠바이 유닛에서 인터페이스를 변경한 후에 마스터/액티브 유닛에서 변경하는 것이 좋습니다. 새 인터페이스는 관리를 위해 다운된 상태로 추가되므로 인터페이스 모니터링에는 영향을 주지 않습니다.

프로시저

단계 1 보안 서비스 모드를 입력합니다.

```
Firepower# scope ssa
```

단계 2 논리적 디바이스를 편집합니다.

```
Firepower /ssa # scope logical-device device_name
```

단계 3 논리적 디바이스에서 인터페이스를 할당 해제합니다.

```
Firepower /ssa/logical-device # delete external-port-link name
```

show external-port-link 명령을 입력하여 인터페이스 이름을 확인합니다.

단계 4 논리적 디바이스에 새 인터페이스를 할당합니다.


```
Firepower /ssa/logical-device* # create external-port-link name interface_id ftd
```

단계 5 구성을 커밋합니다.

```
commit-buffer
```

시스템 구성에 트랜잭션을 커밋합니다.

단계 6 Firepower Management Center에 로그인합니다.

단계 7 **Devices(디바이스) > Device Management(디바이스 관리)**를 선택하고 FTD 디바이스에 대해 수정 아이콘()을 클릭합니다. 기본적으로는 **Interfaces(인터페이스)** 탭이 선택됩니다.

단계 8 **Interfaces(인터페이스)** 탭 왼쪽 상단의 **Sync Interfaces from device(디바이스에서 인터페이스 동기화)** 버튼을 클릭합니다.

단계 9 **Save(저장)**를 클릭합니다.

이제 **Deploy(구축)**를 클릭하고 할당된 디바이스에 정책을 구축할 수 있습니다. 변경사항은 구축할 때까지 활성화되지 않습니다.

ASA 논리적 디바이스의 인터페이스 변경

ASA 논리적 디바이스에서 관리 인터페이스를 할당, 할당 해제 또는 교체할 수 있습니다. ASDM은 새 인터페이스를 자동으로 검색합니다.

시작하기 전에

- 실제 인터페이스 구성, 105 페이지 및 EtherChannel(포트 채널) 추가, 107 페이지에 따라 인터페이스를 구성하고 EtherChannel을 추가합니다.
- 논리적 디바이스에 영향을 주지 않고 할당된 EtherChannel의 멤버십을 수정할 수 있습니다.
- 모든 인터페이스가 기본적으로 클러스터에 할당된 경우와 같이 이미 할당된 인터페이스를 EtherChannel에 추가하려는 경우에는 먼저 논리적 디바이스에서 인터페이스 할당을 해제한 다음 EtherChannel에 인터페이스를 추가해야 합니다. 새 EtherChannel의 경우 이렇게 한 후에 디바이스에 EtherChannel을 할당할 수 있습니다.
- 네트워크 모듈/EtherChannel을 제거하거나 EtherChannel에 할당된 인터페이스를 재할당하는 등 FXOS에서 인터페이스를 제거하면 ASA 구성에서 원래 명령이 유지되므로 필요한 조정을 수행할 수 있습니다. 구성에서 인터페이스를 제거하는 경우에는 구성 전반에 걸쳐 영향을 줄 수 있습니다. ASA OS에서 이전 인터페이스 구성을 수동으로 제거할 수 있습니다.
- 클러스터링 또는 장애 조치의 경우 모든 유닛에서 인터페이스를 추가하거나 제거해야 합니다. 먼저 슬레이브/스탠바이 유닛에서 인터페이스를 변경한 후에 마스터/액티브 유닛에서 변경하는 것이 좋습니다. 새 인터페이스는 관리를 위해 다운된 상태로 추가되므로 인터페이스 모니터링에는 영향을 주지 않습니다.

프로시저

단계 1 보안 서비스 모드를 입력합니다.

```
Firepower# scope ssa
```

단계 2 논리적 디바이스를 편집합니다.

```
Firepower /ssa # scope logical-device device_name
```

단계 3 논리적 디바이스에서 인터페이스를 할당 해제합니다.

```
Firepower /ssa/logical-device # delete external-port-link name
```

show external-port-link 명령을 입력하여 인터페이스 이름을 확인합니다.

관리 인터페이스의 경우 새 관리 인터페이스를 추가하기 전에 현재 인터페이스를 삭제한 다음 **commit-buffer** 명령을 사용하여 변경 사항을 커밋합니다.

단계 4 논리적 디바이스에 새 인터페이스를 할당합니다.

```
Firepower /ssa/logical-device* # create external-port-link name interface_id asa
```

단계 5 구성을 커밋합니다.

commit-buffer

시스템 구성에 트랜잭션을 커밋합니다.

논리적 디바이스 모니터링

• show app

사용 가능한 이미지를 확인합니다.

```
Firepower# scope ssa
Firepower /ssa # show app
```

Name	Version	Author	Supported Deploy Types	CSP Type	Is Default
asa	9.10.1	cisco	Native	Application	Yes
ftd	6.2.3	cisco	Native	Application	Yes

• show app-instance

애플리케이션 인스턴스 상태를 확인합니다.

```
Firepower# scope ssa
Firepower /ssa # show app-instance
```

App Name	Slot ID	Admin State	Oper State	Running Version	Startup Version	Cluster State	Cluster Role
ftd	1	Enabled	Online	6.2.1.62	6.2.1.62	Applicable	None

• show logical-device

논리적 디바이스에 대한 세부사항을 확인합니다.

```
Firepower# scope ssa
Firepower /ssa # show logical-device
```

Logical Device:

Name	Description	Slot ID	Mode	Oper State	Template Name
asa1		1	Standalone	Ok	asa

사이트 간 클러스터링 예시

다음 예에는 지원되는 클러스터 구축에 대한 내용이 나와 있습니다.

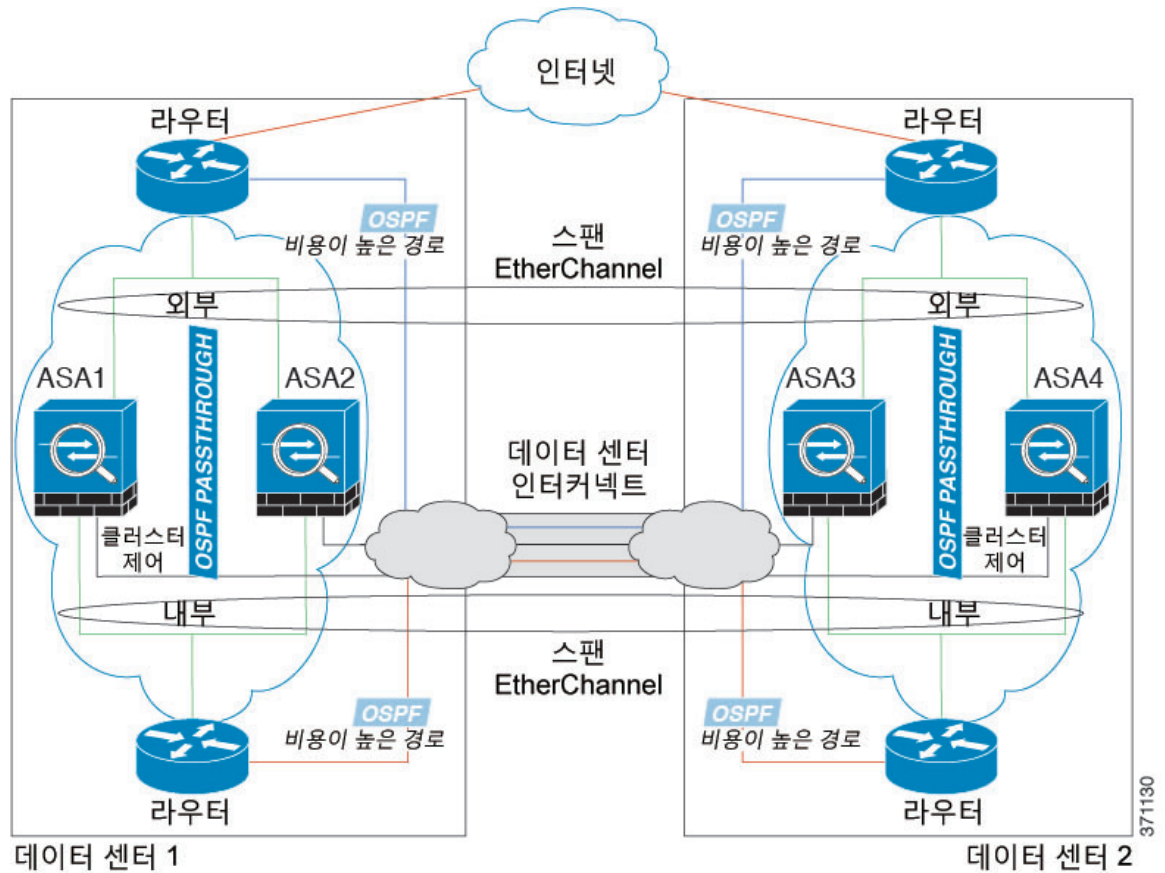
Spanned EtherChannel 투명 모드 노스-사우스 사이트 간의 예

다음 예에서는 내부 라우터와 외부 라우터의 사이에 위치한(노스-사우스 삽입) 2개 데이터 센터 각각에 2개의 클러스터 멤버가 있습니다. 클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부용 스패 EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 새시를 포괄합니다.

각 데이터 센터의 내부 및 외부 라우터에서는 투명 ASA를 통과하는 OSPF를 사용합니다. MAC과 달리 라우터 IP는 모든 라우터마다 고유합니다. DCI를 통해 비용이 높은 경로를 할당하면 특정 사이트의 모든 클러스터 멤버가 가동 중지되지 않는 한 각 데이터 센터 내에서 트래픽이 유지됩니다. ASA를 통과하는 비용이 낮은 경로의 경우, 클러스터의 각 사이트에 있는 같은 브리지 그룹을 거쳐 비대칭 연결을 유지해야 합니다. 어느 한 사이트의 모든 클러스터 멤버에 오류가 발생할 경우, 각 라우터의 트래픽은 DCI를 통해 다른 사이트의 클러스터 멤버로 이동합니다.

각 사이트의 스위치 구현 과정에는 다음 사항이 포함될 수 있습니다.

- 사이트 간 VSS/vPC — 이 시나리오의 경우 데이터 센터 1에 하나의 스위치를 설치하고, 나머지 하나는 데이터 센터 2에 설치합니다. 각 데이터 센터의 클러스터 유닛에 사용할 수 있는 한 가지 옵션은 로컬 스위치에만 연결하는 반면, VSS/vPC 트래픽이 DCI를 통해 통과하도록 하는 것입니다. 이 경우 연결의 대부분은 각 데이터 센터에 로컬로 저장됩니다. DCI에서 추가 트래픽을 처리할 수 있는 경우, 선택에 따라 각 유닛을 DCI 전반의 스위치에 연결할 수 있습니다. 이 경우 트래픽이 데이터 센터 전반에 분산되므로 DCI의 성능이 매우 뛰어나야 합니다.
- 각 사이트의 로컬 VSS/vPC — 스위치 이중화를 개선하기 위해 각 사이트에 별도의 VSS/vPC 쌍을 2개씩 설치할 수 있습니다. 이 경우 여전히 클러스터 유닛의 Spanned EtherChannel은 두 로컬 스위치에만 연결된 데이터 센터 1 새시 및 이러한 로컬 스위치에 연결된 데이터 센터 2 새시로 이루어져 있으나, 사실상 Spanned EtherChannel은 "분리"되어 있습니다. 각 로컬 VSS/vPC에서는 Spanned EtherChannel을 사이트-로컬 EtherChannel로 간주합니다.

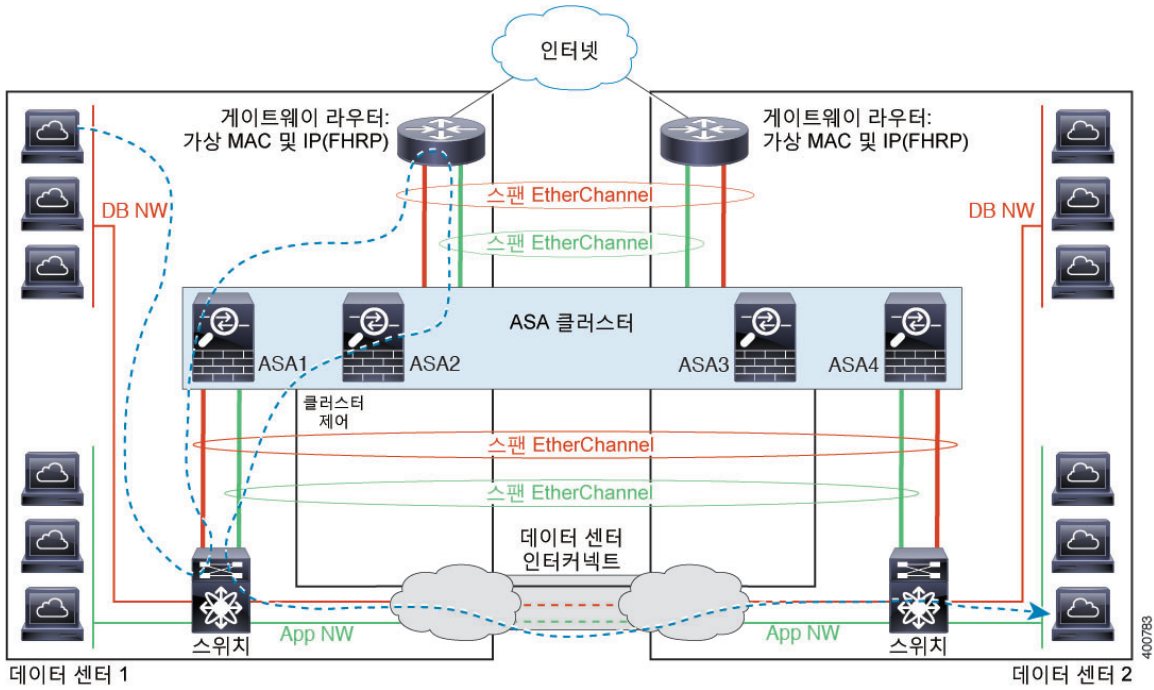


371130

Spanned EtherChannel 투명 모드 이스트-웨스트 사이트 간의 예

다음 예에서는 게이트웨이 라우터와 각 사이트의 두 내부 네트워크, 즉 애플리케이션 네트워크 및 DB 네트워크의 사이에 위치한(이스트-웨스트 삽입) 2개 데이터 센터 각각에 2개의 클러스터 멤버가 있습니다. 클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부에 있는 애플리케이션 및 DB 네트워크에 대한 스팬 EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 새시를 포괄합니다.

각 사이트의 게이트웨이 라우터는 HSRP와 같은 FHRP를 사용하여 각 사이트에 동일한 목적지 가상 MAC 및 IP 주소를 제공합니다. 의도치 않은 MAC 주소 플래핑(flapping)을 피하는 좋은 방법은 `mac-address-table static outside interface mac_address` 명령을 사용하여 게이트웨이 라우터 실제 MAC 주소를 ASA MAC 주소 테이블에 정적으로 추가하는 것입니다. 이러한 항목이 없으면, 사이트 1의 게이트웨이가 사이트 2의 게이트웨이와 통신할 경우 해당 트래픽이 ASA를 통과해 내부 인터페이스에서 사이트 2에 도달하려고 시도하여 문제를 일으킬 수 있습니다. OTV(Overlay Transport Virtualization) 또는 이와 유사한 방법으로 데이터 VLAN이 사이트 전반에 확장됩니다. 트래픽이 게이트웨이 라우터로 예정된 경우 트래픽에서 다른 사이트에 DCI를 전달하는 것을 방지하려면 필터를 추가해야 합니다. 한 개의 사이트에서 게이트웨이 라우터에 연결할 수 없는 경우, 필터를 제거해야 트래픽이 다른 사이트의 게이트웨이 라우터에 전송될 수 있습니다.



vPC/VSS 옵션에 대한 자세한 내용은 [Spanned EtherChannel 투명 모드 노스-사우스 사이트 간의 예, 146 페이지](#)를 참조하십시오.

논리적 디바이스의 기록

기능 이름	플랫폼 릴리스	기능 정보
Firepower 9300에서 ASA 모듈 16개를 위한 인트라 새시 클러스터링(intra-chassis clustering)	1.1.3	현재 ASA를 위한 새시 간 클러스터링을 활성화할 수 있습니다. 최대 16개의 새시에 최대 16개의 모듈을 포함할 수 있습니다.
Firepower 9300에서 ASA를 위한 인트라 새시 클러스터링(intra-chassis clustering)	1.1.1	Firepower 9300 새시 내부에서 모든 ASA 보안 모듈을 클러스터링할 수 있습니다. 추가된 명령: enter cluster-bootstrap, enter logical-device clustered, set chassis-id, set ipv4 gateway, set ipv4 pool, set ipv6 gateway, set ipv6 pool, set key, set mode spanned-etherchannel, set port-type cluster, set service-type, set virtual ipv4, set virtual ipv6



10 장

구성 가져오기/내보내기

- 구성 가져오기/내보내기 정보, 149 페이지
- FXOS 구성 파일 내보내기, 150 페이지
- 자동 구성 내보내기 예약, 152 페이지
- 구성 내보내기 미리 알림 설정, 153 페이지
- 구성 파일 가져오기, 154 페이지

구성 가져오기/내보내기 정보

구성 내보내기 기능을 사용하여 Firepower 9300 새시의 논리적 디바이스 및 플랫폼 구성 설정을 포함하는 XML 파일을 원격 서버로 내보낼 수 있습니다. 나중에 해당 구성 파일을 가져와서 구성 설정을 Firepower 9300 새시에 빠르게 적용하여, 알려진 정상적인 구성으로 돌아가거나 시스템 장애로부터 복구할 수 있습니다.

지침 및 제한 사항

- 구성 파일의 내용을 수정하지 마십시오. 구성 파일을 수정하면 해당 파일을 사용한 구성 가져오기가 실패할 수 있습니다.
- 애플리케이션 관련 구성 설정은 구성 파일에 포함되지 않습니다. 애플리케이션 관련 설정 및 구성을 관리하려면 애플리케이션에서 제공하는 구성 백업 도구를 사용해야 합니다.
- Firepower 9300 새시에서 구성을 가져오면 Firepower 9300 새시에 있는 모든 기존의 구성(논리적 디바이스 포함)이 삭제되고 가져오기 파일에 포함된 구성으로 완전히 교체됩니다.
- 구성을 가져올 경우 원래 구성을 내보낸 동일한 Firepower 9300 새시로만 가져오는 것이 좋습니다.
- 구성을 가져오는 Firepower 9300 새시의 플랫폼 소프트웨어 버전은 내보낼 때와 동일한 버전이어야 합니다. 버전이 다르면 가져오기 작업의 성공이 보장되지 않습니다. Firepower 9300 새시를 업그레이드 또는 다운그레이드할 때마다 백업 구성을 내보내는 것이 좋습니다.
- 구성을 가져오는 Firepower 9300 새시에는 내보냈을 때와 동일한 슬롯에 동일한 네트워크 모듈이 설치되어 있어야 합니다.

- 구성을 가져오는 Firepower 9300 새시에는, 가져오는 내보내기 파일에 정의된 논리적 디바이스에 대해 올바른 소프트웨어 애플리케이션 이미지가 설치되어 있어야 합니다.
- 기존 백업 파일을 덮어쓰지 않으려면 백업 작업 시 파일 이름을 변경하거나 기존 파일을 다른 위치에 복사하십시오.

FXOS 구성 파일 내보내기

Firepower 9300 새시의 논리적 디바이스 및 플랫폼 구성 설정을 포함하는 XML 파일을 원격 서버로 내보내려면 구성 내보내기 기능을 사용합니다.

구성 내보내기 기능 사용에 대한 중요한 정보는 [구성 가져오기/내보내기 정보](#)를 참조하십시오.

프로시저

단계 1 원격 서버로 구성 파일을 내보내려면 다음을 수행합니다.

scope system

export-config *URL* **enabled** **commit-buffer**

다음 구문 중 하나를 사용하여 내보낼 파일의 URL을 지정합니다.

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

참고 파일 이름을 포함한 전체 경로를 지정해야 합니다. 파일 이름을 지정하지 않으면 지정된 경로에 숨김 파일이 생성됩니다.

예제:

```
Firepower-chassis# scope system
Firepower-chassis /system # export-config scp://user1@192.168.1.2:/export/cfg-backup.xml
enabled
Firepower-chassis /system/export-config # commit-buffer
```

단계 2 내보내기 작업의 상태를 확인하려면:

scope system

scope export-config *hostname*

show fsm status

예제:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope export-config 192.168.1.2
```



```
Firepower-chassis /system/export-config # show fsm status

Hostname: 192.168.1.2

FSM 1:
  Remote Result: Not Applicable
  Remote Error Code: None
  Remote Error Description:
  Status: Nop
  Previous Status: Backup Success
  Timestamp: 2016-01-03T15:32:08.636
  Try: 0
  Progress (%): 100
  Current Task:
```

단계 3 기존의 내보내기 작업을 보려면 다음을 수행합니다.

```
scope system
```

```
show export-config
```

단계 4 기존 내보내기 작업을 수정하려면 다음을 수행합니다.

```
scope system
```

```
scope export-config hostname
```

내보내기 작업을 수정하려면 다음 명령을 사용합니다.

- **{enable|disable}**
- **set description** *<description>*
- **set password** *<password>*
- **set port** *<port>*
- **set protocol** {ftp|scp|sftp|tftp}
- **set remote-file** *path_and_filename*
- **set user** *<user>*

단계 5 내보내기 작업을 삭제하려면 다음을 수행합니다.

```
scope system
```

```
delete export-config hostname
```

```
commit-buffer
```

자동 구성 내보내기 예약

Firepower 9300 새시의 논리적 디바이스 및 플랫폼 구성 설정을 포함하는 XML 파일을 원격 서버로 자동으로 내보내려면 예약된 내보내기 기능을 사용합니다. 내보내기를 매일, 매주 또는 2주마다 실행하도록 예약할 수 있습니다. 구성 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.

구성 내보내기 기능 사용에 대한 중요한 정보는 [구성 가져오기/내보내기 정보](#)를 참조하십시오.

프로시저

예약된 내보내기 작업을 생성하려면:

- 정책 구성을 내보낼 범위를 설정합니다.

scope org

scope cfg-export-policy default

- 내보내기 정책을 활성화합니다.

set adminstate enable

- 원격 서버와의 통신에서 사용할 프로토콜을 지정합니다.

set protocol {ftp|scp|sftp|tftp}

- 백업 파일을 저장할 위치의 IP 주소 또는 호스트 이름을 지정합니다. 이는 Firepower 9300 새시가 네트워크를 통해 액세스할 수 있는 서버, 스토리지 어레이, 로컬 드라이브, 기타 읽기/쓰기 미디어일 수 있습니다.

IP 주소가 아니라 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다.

set hostname hostname

- 기본값 이외의 포트를 사용하는 경우 포트 번호를 지정합니다.

set port port

- 원격 서버에 로그인할 때 사용할 사용자 이름을 지정합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.

set user username

- 원격 서버 사용자 이름의 비밀번호를 지정합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.

set password password

- 구성 파일을 내보낼 전체 경로(파일 이름 포함)를 지정합니다. 파일 이름을 생략할 경우 내보내기 절차에서 파일에 이름을 할당합니다.

set remote-file path_and_filename

- i) 구성 자동 내보내기를 수행할 일정을 지정합니다. Daily(매일), Weekly(매주) 또는 BiWeekly(격주) 중 하나일 수 있습니다.

set schedule {daily|weekly|bi-weekly}

- j) 시스템 구성에 트랜잭션을 커밋합니다.

commit-buffer

예제:

```
Firepower-chassis# scope org
Firepower-chassis /org # scope cfg-export-policy default
Firepower-chassis /org/cfg-export-policy # set adminstate enable
Firepower-chassis /org/cfg-export-policy* # set protocol scp
Firepower-chassis /org/cfg-export-policy* # set hostname 192.168.1.2
Firepower-chassis /org/cfg-export-policy* # set remote-file /export/cfg-backup.xml
Firepower-chassis /org/cfg-export-policy* # set user user1
Firepower-chassis /org/cfg-export-policy* # set password
Password:
Firepower-chassis /org/cfg-export-policy* # set schedule weekly
Firepower-chassis /org/cfg-export-policy* # commit-buffer
Firepower-chassis /org/cfg-export-policy #
Firepower-chassis /org/cfg-export-policy # show detail
```

```
Config Export policy:
  Name: default
  Description: Configuration Export Policy
  Admin State: Enable
  Protocol: Scp
  Hostname: 192.168.1.2
  User: user1
  Remote File: /export/cfg-backup.xml
  Schedule: Weekly
  Port: Default
  Current Task:
```

구성 내보내기 미리 알림 설정

특정 기간(일수)에 구성 내보내기가 실행되지 않은 경우 시스템에서 오류를 생성하도록 하려면 Export Reminder(내보내기 미리 알림) 기능을 사용합니다.

프로시저

구성 내보내기 미리 알림을 생성하려면 다음을 수행합니다.

scope org

scope cfg-export-reminder

set frequency days

set adminstate {enable|disable}

commit-buffer

예제:

```

Firepower-chassis# scope org
Firepower-chassis /org # scope cfg-export-reminder
Firepower-chassis /org/cfg-export-reminder # set frequency 10
Firepower-chassis /org/cfg-export-reminder* # set adminstate enable
Firepower-chassis /org/cfg-export-reminder* # commit-buffer
Firepower-chassis /org/cfg-export-reminder # show detail

Config Export Reminder:
  Config Export Reminder (Days): 10
  AdminState: Enable

```

구성 파일 가져오기

Firepower 9300 새시에서 전에 내보낸 구성 설정을 적용하려면 구성 가져오기 기능을 사용할 수 있습니다. 이 기능을 사용하면 알려진 양호한 구성으로 돌아가거나 시스템 장애로부터 복구할 수 있습니다. 구성 가져오기 기능 사용에 대한 중요한 정보는 [구성 가져오기/내보내기 정보](#)를 참조하십시오.

프로시저

단계 1 원격 서버에서 구성 파일을 가져오려면 다음을 수행합니다.

scope system**import-config** *URL* **enabled****commit-buffer**

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

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

예제:

```

Firepower-chassis# scope system
Firepower-chassis /system # import-config scp://user1@192.168.1.2:/import/cfg-backup.xml
enabled
Warning: After configuration import any changes on the breakout port configuration will
cause the system to reboot
Firepower-chassis /system/import-config # commit-buffer

```

단계 2 가져오기 작업의 상태를 확인하려면 다음을 수행합니다.

scope system**scope import-config** *hostname***show fsm status**

예제:

```

Firepower-chassis# scope system
Firepower-chassis /system # scope import-config 192.168.1.2
Firepower-chassis /system/import-config # show fsm status

Hostname: 192.168.1.2

FSM 1:
  Remote Result: Not Applicable
  Remote Error Code: None
  Remote Error Description:
  Status: Import Wait For Switch
  Previous Status: Import Config Breakout
  Timestamp: 2016-01-03T15:45:03.963
  Try: 0
  Progress (%): 97
  Current Task: updating breakout port configuration(FSM-STAGE:sam:dme:
    MgmtImporterImport:configBreakout)

```

단계 3 기존 가져오기 작업을 보려면 다음을 수행합니다.

scope system**show import-config**

단계 4 기존 가져오기 작업을 수정하려면 다음을 수행합니다.

scope system**scope import-config** *hostname*

가져오기 작업을 수정하려면 다음 명령을 사용합니다.

- **{enable|disable}**
- **set description** *<description>*
- **set password** *<password>*
- **set port** *<port>*
- **set protocol** {ftp|scp|sftp|tftp}
- **set remote-file** *path_and_filename*
- **set user** *<user>*

단계 5 가져오기 작업을 삭제하려면 다음을 수행합니다.

scope system**delete import-config** *hostname*

commit-buffer



11 장

패킷 캡처

- 패킷 캡처, 157 페이지
- 패킷 캡처 관련 지침 및 제한 사항, 158 페이지
- 패킷 캡처 세션 생성 또는 수정, 158 페이지
- 패킷 캡처에 대한 필터 구성, 160 페이지
- 패킷 캡처 세션 시작 및 중지, 162 페이지
- 패킷 캡처 파일 다운로드, 162 페이지

패킷 캡처

패킷 캡처는 연결 및 구성 문제를 디버깅하고 Firepower 9300 새시를 통과하는 트래픽 흐름을 파악하기 위해 사용할 수 있는 매우 유용한 자산입니다. 패킷 캡처 도구를 사용하면 Firepower 9300 새시의 특정 인터페이스를 통과하는 트래픽을 로깅할 수 있습니다.

여러 패킷 캡처 세션을 생성할 수 있으며, 각 세션은 여러 인터페이스의 트래픽을 캡처할 수 있습니다. 패킷 캡처 세션에 포함된 각 인터페이스에 대해 별도의 패킷 캡처(PCAP) 파일이 생성됩니다.

백플레인 포트 매핑

Firepower 9300 새시는 내부 백플레인 포트에 다음 매핑을 사용합니다.

보안 모듈	포트 매핑	설명
보안 모듈 1/보안 엔진	Ethernet1/9	Internal-Data0/0
보안 모듈 1/보안 엔진	Ethernet1/10	Internal-Data1/0
보안 모듈 2	Ethernet1/11	Internal-Data0/0
보안 모듈 2	Ethernet1/12	Internal-Data1/0
보안 모듈 3	Ethernet1/13	Internal-Data0/0
보안 모듈 3	Ethernet1/14	Internal-Data1/0

패킷 캡처 관련 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.
- 소스 또는 목적지 IPv6 주소를 기반으로 필터링할 수 있는 옵션이 없습니다.
- 내부 스위치의 인그레스 단계에서만 캡처합니다.
- 내부 스위치에서 이해할 수 없는 패킷(Security Group Tag 및 Network Service Header 패킷)에는 필터가 효과적이지 않습니다.
- EtherChannel 전체나, 그러나 논리적 디바이스에 할당된 EtherChannel의 경우에는 EtherChannel의 각 멤버 인터페이스에서 패킷을 캡처할 수 있습니다.
- 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

프로시저

단계 1 패킷 캡처 모드로 들어갑니다.

```
Firepower-chassis # scope packet-capture
```

단계 2 필터를 생성합니다. [패킷 캡처에 대한 필터 구성, 160 페이지](#) 섹션을 참조하십시오.

패킷 캡처 세션에 포함된 인터페이스에 필터를 적용할 수 있습니다.

단계 3 패킷 캡처 세션을 생성하거나 수정하려면 다음을 수행합니다.

```
Firepower-chassis /packet-capture # enter session session_name
```

단계 4 이 패킷 캡처 세션을 사용할 버퍼 크기를 지정합니다.

```
Firepower-chassis /packet-capture/session* # set session-memory-usage session_size_in_megabytes
```

256~2048MB 범위에서 버퍼 크기를 지정해야 합니다.

단계 5 이 패킷 캡처 세션에 포함해야 할 물리적 소스 포트를 지정합니다.

여러 포트에서 캡처할 수 있으며, 동일한 패킷 캡처 세션 중에 물리적 포트와 애플리케이션 포트 둘다에서 캡처할 수 있습니다. 세션에 포함된 각 포트에 대해 별도의 패킷 캡처 파일이 생성됩니다. EtherChannel 전체에 대해 패킷을 캡처할 수는 없습니다. 그러나 논리적 디바이스에 할당된 EtherChannel의 경우에는 EtherChannel의 각 멤버 인터페이스에서 패킷을 캡처할 수 있습니다.

참고 패킷 캡처 세션에서 포트를 제거하려면 아래에 나열된 명령에서 **create** 대신 **delete**를 사용합니다.

- a) 물리적 포트를 지정합니다.

```
Firepower-chassis /packet-capture/session* # create {phy-port | phy-aggr-port} port_id
```

예제:

```
Firepower-chassis /packet-capture/session* # create phy-port Ethernet1/1
Firepower-chassis /packet-capture/session/phy-port* #
```

- b) (선택 사항) 원하는 필터를 적용합니다.

```
Firepower-chassis /packet-capture/session/phy-port* # set {source-filter} filtername
```

참고 포트에서 필터를 제거하려면 **set source-filter ""**를 사용합니다.

- c) 위의 단계를 필요한 만큼 반복하여 원하는 모든 포트를 추가합니다.

단계 6 이 패킷 캡처 세션에 포함해야 할 애플리케이션 소스 포트를 지정합니다.

여러 포트에서 캡처할 수 있으며, 동일한 패킷 캡처 세션 중에 물리적 포트와 애플리케이션 포트 둘다에서 캡처할 수 있습니다. 세션에 포함된 각 포트에 대해 별도의 패킷 캡처 파일이 생성됩니다.

참고 패킷 캡처 세션에서 포트를 제거하려면 아래에 나열된 명령에서 **create** 대신 **delete**를 사용합니다.

- a) 애플리케이션 포트를 지정합니다.

```
Firepower-chassis /packet-capture/session* # create app_port module_slot link_name interface_name
app_name
```

- b) (선택 사항) 원하는 필터를 적용합니다.

```
Firepower-chassis /packet-capture/session/phy-port* # set {source-filter} filtername
```

참고 포트에서 필터를 제거하려면 **set source-filter ""**를 사용합니다.

- c) 위의 단계를 필요한 만큼 반복하여 원하는 모든 애플리케이션 포트를 추가합니다.

단계 7 패킷 캡처 세션을 지금 시작하려면:

```
Firepower-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 구성에 트랜잭션을 커밋합니다.

```
Firepower-chassis /packet-capture/session* # commit-buffer
```

패킷 캡처 세션을 활성화하면 시스템에서 패킷 캡처를 시작합니다. 세션에서 PCAP 파일을 다운로드하려면 먼저 캡처를 중지해야 합니다.

예

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # create session asalinside
Firepower-chassis packet-capture/session # set session-memory-usage 256
Firepower-chassis packet-capture/session* # create phy-port Ethernet3/1
Firepower-chassis packet-capture/session* # create phy-aggr-port Ethernet2/1/1
Firepower-chassis packet-capture/session* # create app-port 1 link1 Ethernet 1/1 asa
Firepower-chassis packet-capture/session* # create filter interfacelvlan100
Firepower-chassis packet-capture/filter* # set ivlan 100
Firepower-chassis packet-capture/filter* # set srcIP 6.6.6.6
Firepower-chassis packet-capture/filter* # set srcPort 80
Firepower-chassis packet-capture/filter* # set destIP 10.10.10.10
Firepower-chassis packet-capture/filter* # set destPort 5050
Firepower-chassis packet-capture/filter* # exit
Firepower-chassis packet-capture/session* # scope phy-port Ethernet3/1
Firepower-chassis packet-capture/session/phy-port* # set src-filter interfacelvlan100
Firepower-chassis packet-capture/session/phy-port* # exit
Firepower-chassis packet-capture/session* # scope app-port 1 link1 Ethernet1/1 asa
Firepower-chassis packet-capture/session/app-port* # set src-filter interfacelvlan100
Firepower-chassis packet-capture/session/app-port* # exit
Firepower-chassis packet-capture/session* # enable
Firepower-chassis packet-capture/session* # commit-buffer
Firepower-chassis packet-capture/session #
```

패킷 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.

프로시저

단계 1 패킷 캡처 모드로 들어갑니다.

```
Firepower-chassis # scope packet-capture
```

단계 2 새 패킷 캡처 필터를 생성하려면:

```
Firepower-chassis /packet-capture # create filter filter_name
```

기존의 패킷 캡처 필터를 수정하려면:

```
Firepower-chassis /packet-capture # enter filter filter_name
```

기존의 패킷 캡처 필터를 삭제하려면:

```
Firepower-chassis /packet-capture # delete filter filter_name
```

단계 3 하나 이상의 필터 속성을 설정하여 필터 세부 사항을 지정합니다.

```
Firepower-chassis /packet-capture/filter* # set <filterprop filterprop_value
```

표 5: 지원되는 필터 속성

ivlan	Inner VLAN ID(포트로 들어가는 동안 패킷의 vlan)
ovlan	Outer VLAN ID(Firepower 9300 새시에 의해 추가된 vlan)
srcip	소스 IP 주소(IPv4)
destip	목적지 IP 주소(IPv4)
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

예

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # create filter interfacelvlan100
Firepower-chassis packet-capture/filter* # set ivlan 100
Firepower-chassis packet-capture/filter* # set srcip 6.6.6.6
Firepower-chassis packet-capture/filter* # set srcport 80
Firepower-chassis packet-capture/filter* # set destip 10.10.10.10
Firepower-chassis packet-capture/filter* # set destport 5050
Firepower-chassis packet-capture/filter* # commit-buffer
```

패킷 캡처 세션 시작 및 중지

프로시저

단계 1 패킷 캡처 모드로 들어갑니다.

```
Firepower-chassis # scope packet-capture
```

단계 2 시작하거나 중지할 패킷 캡처 세션의 범위를 입력합니다.

```
Firepower-chassis /packet-capture # enter session session_name
```

단계 3 패킷 캡처 세션을 시작하려면:

```
Firepower-chassis /packet-capture/session* # enable [append | overwrite]
```

참고 다른 세션이 실행 중인 동안에는 패킷 캡처 세션을 시작할 수 없습니다.

패킷 캡처 세션이 실행 중인 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:

```
Firepower-chassis /packet-capture/session* # disable
```

단계 5 시스템 구성에 트랜잭션을 커밋합니다.

```
Firepower-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.

예

```
Firepower-chassis# scope packet-capture
Firepower-chassis packet-capture # scope session asalinside
Firepower-chassis packet-capture/session # enable append
Firepower-chassis packet-capture/session* # commit-buffer
Firepower-chassis packet-capture/session #
```

패킷 캡처 파일 다운로드

네트워크 패킷 분석기를 사용하여 분석할 수 있도록 세션에서 로컬 컴퓨터로 PCAP(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
```

프로시저

Firepower 9300 새시에서 PCAP 파일을 복사하려면:

참고 패킷 캡처 세션에서 PCAP 파일을 다운로드하려면 먼저 해당 세션을 중지해야 합니다.

a) 로컬 관리에 연결합니다.

```
Firepower-chassis # connect localmgmt
```

b) PCAP 파일을 복사합니다.

```
# copy pcap_file copy_destination
```

예

```
Firepower-chassis# connect localmgmt  
# copy workspace://packet-capture/session-1/test-ethernet-1-1-0.pcap  
scp://user@10.10.10.1:/workspace/
```




색인

ㄱ

- 객체 명령 **5**
- 관리 객체 **3**
- 관리 IP 주소 **57**
 - 변경 **57**
- 구성 **75, 76, 77, 78, 80, 81**
 - HTTPS **75, 76, 77, 78, 80, 81**
- 구성 가져오기 **149**
- 구성 가져오기/내보내기 **149**
 - 제한 사항 **149**
 - 지침 **149**
- 구성 내보내기 **149**
- 기록, 비밀번호 **31**

ㄴ

- 날짜 **63**
 - 수동으로 설정 **63**
- 날짜 및 시간 **59**
 - 구성 **59**
- 논리적 디바이스 **51, 116, 120, 131, 138, 140**
 - 독립형 생성 **120**
 - 삭제 **140**
 - 연결 **138**
 - 연결 종료 **138**
 - 이미지 버전 업데이트 **51**
 - 클러스터 생성 **116, 131**
- 논리적 디바이스 연결 종료 **138**
- 논리적 디바이스에 연결 **138**
- 높은 수준의 작업 목록 **9**

D

- DNS **100**

ㅍ

- 명령 **6**
 - history **6**
- 명령 모드 **3**

ㅂ

- 보류 중인 명령 **6**
- 비밀번호 **27, 31, 36**
 - 기록 수 **31**
 - 변경 간격 **31**
 - 보안 수준 확인 **36**
 - 지침 **27**
- 비밀번호 보안 수준 적용 **36**
- 비밀번호 프로파일 **31, 37, 38, 39, 43**
 - 변경 간격 **37**
 - 변경 안 함 간격 **38**
 - 비밀번호 기록 수 **39**
 - 비밀번호 기록 지우기 **43**
 - 정보 **31**

ㅅ

- 사용 **69**
 - SNMP **69**
- 사용자 **26, 27, 31, 32, 37, 38, 39, 43**
 - 기본 인증 **32**
 - 로컬로 인증 **31, 37, 38, 39, 43**
 - 명명 지침 **26**
 - 비밀번호 지침 **27**
 - 역할 **31**
- 사용자 계정 **37, 38, 39, 43**
 - 비밀번호 프로파일 **37, 38, 39, 43**
- 사용자 어카운트 **31**
 - 비밀번호 프로파일 **31**
- 새시 **9**
 - 초기 구성 **9**
- 세션 시간 초과 **34**
- 시간 **63**
 - 수동으로 설정 **63**

ㅇ

- 알림 **67**
 - 정보 **67**

어카운트 **31, 37, 38, 39, 43**

로컬로 인증 **31, 37, 38, 39, 43**

원격 사용자의 역할 정책 **35**

위협 방어 이미지 **49**

Firepower Security Appliance에 다운로드 **49**

이미지 **45, 46, 47, 48, 49**

관리 **45**

무결성 확인 **47**

Cisco.com에서 다운로드 **46**

Firepower eXtensible 운영 체제 플랫폼 번들 업그레이드 **48**

Firepower Security Appliance에 다운로드 **46, 49**

이미지 버전 **51**

업데이트 **51**

인증 **32**

기본 **32**

인증서 **75**

정보 **75**

인터페이스 **105**

구성 **105**

속성 **105**

ㄷ

작업 흐름 **9**

정책 **35**

원격 사용자의 역할 **35**

ㄸ

초기 구성 **9**

ㄷ

커뮤니티, SNMP **69**

콘솔 **34**

timeout **34**

클러스터 **116, 126, 131**

생성 **116, 131**

정보 **126**

클러스터링 **118, 127, 128, 129**

관리 **129**

network **129**

클러스터 제어 링크 **127, 128**

redundancy **128**

size **127**

device-local EtherChannels, 스위치에서 구성 **118**

키 링 **75, 76, 77, 78, 80, 81, 85**

삭제 **85**

생성 **75**

인증서 가져오기 **81**

인증서 요청 **77, 78**

키 링 (계속)

재생성 **76**

정보 **75**

트러스트 포인트 **80**

ㅈ

통신 서비스 **69, 75, 76, 77, 78, 80, 81**

HTTPS **75, 76, 77, 78, 80, 81**

SNMP **69**

트랩 **67, 70, 71**

삭제 **71**

생성 **70**

정보 **67**

트러스트 포인트 **75, 80, 86**

삭제 **86**

생성 **80**

정보 **75**

ㅊ

패킷 캡처 **157, 158, 160, 162**

패킷 캡처 세션 생성 **158**

패킷 캡처 세션 시작 **162**

패킷 캡처 세션 중지 **162**

필터 **160**

PCAP 파일 다운로드 **162**

패킷 캡처 세션 생성 **158**

패킷 캡처 파일 다운로드 **162**

펌웨어 **53**

업그레이드 **53**

펌웨어 업그레이드 **53**

포트 채널 **107**

구성 **107**

표준 시간대 **59, 61, 63**

설정 **59, 61, 63**

프로파일 **31**

비밀번호 **31**

플랫폼 번들 **45, 46, 47, 48**

무결성 확인 **47**

업그레이드 **48**

정보 **45**

Cisco.com에서 다운로드 **46**

Firepower Security Appliance에 다운로드 **46**

A

AAA **88, 89, 92, 93, 94, 95, 96, 98**

LDAP 제공자 **88, 89, 92**

RADIUS 제공자 **93, 94, 95**

TACACS+ 제공자 **96, 98**

asa **51, 116, 120, 131, 138, 140**
 논리적 디바이스 삭제 **140**
 독립형 ASA 논리적 디바이스 생성 **120**
 연결 **138**
 연결 종료 **138**
 이미지 버전 업데이트 **51**
 클러스터 생성 **116, 131**
 ASA 이미지 **45, 46, 49**
 정보 **45**
 Cisco.com에서 다운로드 **46**
 Firepower Security Appliance에 다운로드 **49**
 authNoPriv **67**
 authPriv **67**
 Breakout 케이블 **109**
 구성 **109**
 Breakout 포트 **109**
 call home **20**
 HTTP 프록시 구성 **20**
 Cisco Secure Package **45, 46, 49**
 정보 **45**
 Cisco.com에서 다운로드 **46**
 Firepower Security Appliance에 다운로드 **49**
 CLI, 참조 (Command Line Interface)
 CLI 세션 제한 **7**
 CLI(Command Line Interface) **12**
 액세스 **12**
 CLI(Command Line Interface) 액세스 **12**
 clustering **114, 116**
 멤버 요구 사항 **114**
 소프트웨어 업그레이드 **114**
 소프트웨어 요구 사항 **114**
 spanning-tree portfast **116**
 CSP, 참조 Cisco Secure Package
 Firepower 새시 **9**
 초기 구성 **9**
 Firepower 플랫폼 번들 **45, 46, 47, 48**
 무결성 확인 **47**
 업그레이드 **48**
 정보 **45**
 Cisco.com에서 다운로드 **46**
 Firepower Security Appliance에 다운로드 **46**
 Firepower eXtensible OS **48**
 플랫폼 번들 업그레이드 **48**
 Firepower Security Appliance **1**
 개요 **1**
 fpga **53**
 업그레이드 **53**
 HTTP 프록시 **20**
 구성 **20**
 HTTPS **34, 75, 76, 77, 78, 80, 81, 83, 84, 86**
 구성 **83**

HTTPS (계속)
 비활성화 **86**
 인증서 가져오기 **81**
 인증서 요청 **77, 78**
 키 링 생성 **75**
 키 링 재생성 **76**
 트러스트 포인트 **80**
 포트 변경 **84**
 timeout **34**
 LDAP **88, 89, 92**
 LDAP 제공자 **89, 92**
 삭제 **92**
 생성 **89**
 License Authority **21**
 noAuthNoPriv **67**
 NTP **59, 61, 62**
 구성 **59, 61**
 삭제 **62**
 추가 **61**

P

PCAP, 참조 패킷 캡처
 PCAP 파일 **162**
 다운로드 **162**
 PKI **75**

R

RADIUS **93, 94, 95**
 RADIUS 제공자 **94, 95**
 삭제 **95**
 생성 **94**
 rommon **53**
 업그레이드 **53**
 RSA **75**

S

Smart Call Home **20**
 HTTP 프록시 구성 **20**
 SNMP **66, 67, 68, 69, 70, 71, 72, 74**
 권한 **67**
 버전 3 보안 기능 **68**
 보안 수준 **67**
 사용 **69**
 사용자 **72, 74**
 삭제 **74**
 생성 **72**
 알림 **67**
 정보 **66**
 지원 **66, 68**

SNMP (계속)

- 커뮤니티 **69**
- 트랩 **70, 71**
 - 삭제 **71**
 - 생성 **70**

SNMPv3 **68**

- 보안 기능 **68**

SSH **34, 64**

- 구성 **64**
- timeout **34**

syslog **98**

- 로컬 대상 구성 **98**
- 로컬 소스 구성 **98**
- 원격 대상 구성 **98**

system **9**

- 초기 구성 **9**

TTACACS+ **96, 98**TACACS+ 제공자 **96, 98**

- 삭제 **98**

TACACS+ 제공자 (계속)

- 생성 **96**

Telnet **34, 65**

- 구성 **65**
- timeout **34**

timeout **34**

- 콘솔 **34**

HTTPS, SSH 및 텔넷 **34****U**users **7, 25, 35, 36, 40, 42, 43, 72, 74**

- 관리 **25**

비밀번호 보안 수준 확인 **36**

- 비활성화 **43**

- 삭제 **42**

- 생성 **40**

- 원격, 역할 정책 **35**

- 활성화 **43**

- CLI 세션 제한 **7**

- SNMP **72, 74**