

# FTD(Firepower 위협 방어) 클러스터 문제 해결

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[클러스터 기본 사항](#)

[NGFW 아키텍처](#)

[클러스터 캡처](#)

[CCL\(Cluster Control Link\) 메시지](#)

[CCP\(Cluster Control Point\) 메시지](#)

[HC\(Cluster Health-Check\) 메커니즘](#)

[클러스터 HC 오류 시나리오](#)

[클러스터 데이터 플레인 연결 설정](#)

[문제 해결](#)

[클러스터 문제 해결 소개](#)

[클러스터 데이터 플레인 문제](#)

[NAT/PAT 공통 문제](#)

[프래그먼트 처리](#)

[ACI 문제](#)

[클러스터 컨트롤 플레인 문제](#)

[유닛이 클러스터에 참가할 수 없음](#)

[CCL의 MTU 크기](#)

[클러스터 유닛 간의 인터페이스 불일치](#)

[데이터/포트 채널 인터페이스 문제](#)

[CCL에 대한 도달 가능성 문제로 인한 스플릿 브레인](#)

[일시 중단된 데이터 포트 채널 인터페이스로 인해 클러스터가 비활성화되었습니다.](#)

[클러스터 안정성 문제](#)

[FXOS 역추적](#)

[디스크 가득 참](#)

[오버플로 보호](#)

[간소화된 모드](#)

[관련 정보](#)

---

## 소개

이 문서에서는 NGFW(Firepower Next-Generation Firewall)에서 클러스터 설정을 해결하는 방법에 대해 설명합니다.

## 사전 요구 사항

# 요구 사항

Cisco에서는 다음 항목에 대해 알고 있는 것이 좋습니다(링크는 관련 정보 섹션 참조).

- Firepower 플랫폼 아키텍처
- Firepower 클러스터 컨피그레이션 및 운영
- FTD 및 FXOS(Firepower eXtensible Operating System) CLI 속지
- NGFW/데이터 플레인 로그
- NGFW/데이터 플레인 패킷 추적기
- FXOS/데이터 플레인 캡처

## 사용되는 구성 요소

- 하드웨어: Firepower 4125
- 소프트웨어: 6.7.0(빌드 65) - 데이터 플레인 9.15(1)

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 배경 정보

이 문서에서 다루는 대부분의 항목은 ASA(Adaptive Security Appliance) 클러스터 트러블슈팅에도 완전하게 적용됩니다.

## 구성

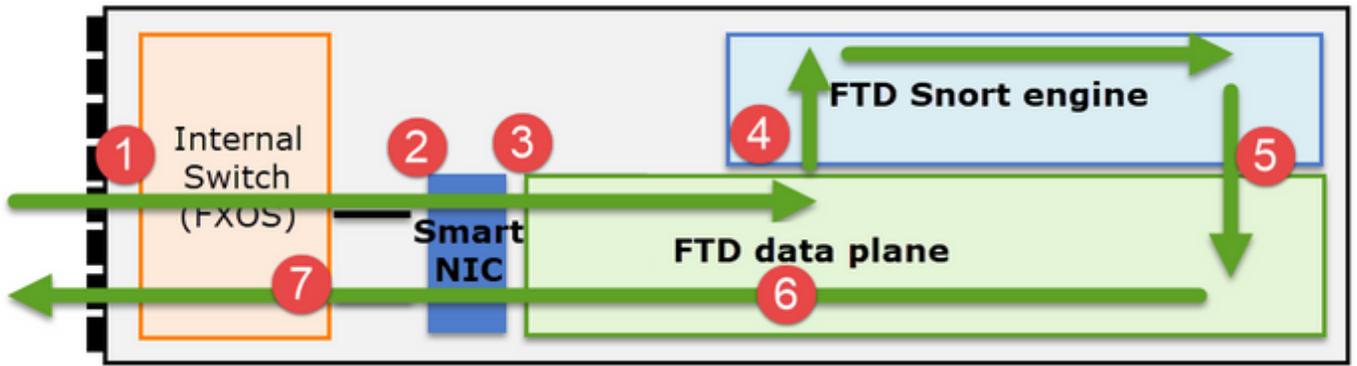
클러스터 구축의 컨피그레이션 부분은 FMC 및 FXOS 컨피그레이션 가이드에서 다룹니다.

- [firepower 위협 방어를 위한 클러스터링](#)
- [확장성 및 고가용성을 위한 Firepower 위협 방어를 위한 클러스터 구축](#)

## 클러스터 기본 사항

### NGFW 아키텍처

firepower 41xx 또는 93xx 시리즈에서 전송 패킷을 처리하는 방법을 이해하는 것이 중요합니다.



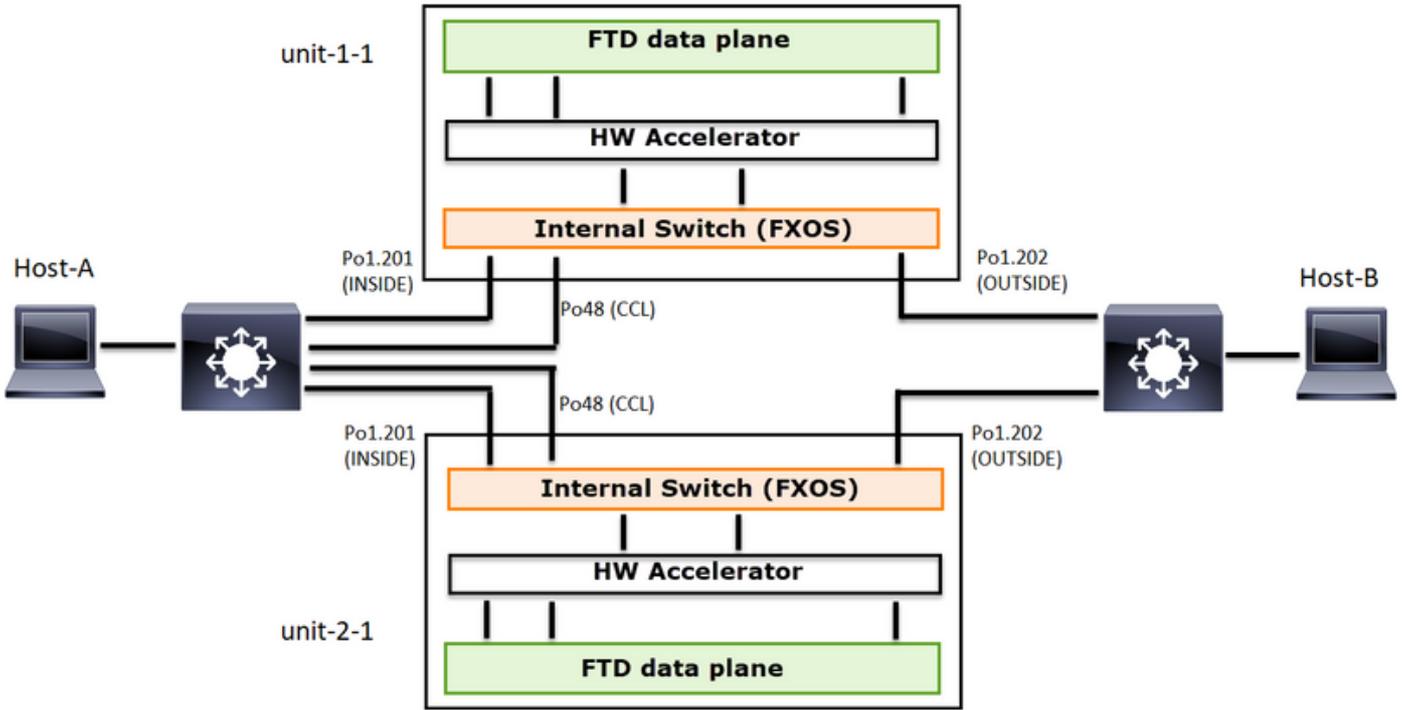
1. 패킷은 인그레스 인터페이스로 들어가며 새시 내부 스위치에 의해 처리됩니다.
2. 패킷이 Smart NIC를 통과합니다. 플로우가 오프로드된 경우(HW 가속), 패킷은 Smart NIC에 의해서만 처리된 다음 네트워크로 다시 전송됩니다.
3. 패킷이 오프로드되지 않은 경우, 주로 L3/L4 검사를 수행하는 FTD 데이터 플레인으로 들어갑니다.
4. 정책에 필요한 경우 Snort 엔진에서 패킷을 검사합니다(주로 L7 검사).
5. Snort 엔진은 패킷에 대한 판정(예: 허용 또는 차단)을 반환합니다.
6. 데이터 평면은 Snort의 판정을 기반으로 패킷을 삭제하거나 전달합니다.
7. 패킷이 내부 새시 스위치를 통해 새시를 이그레스(egress)합니다.

## 클러스터 캡처

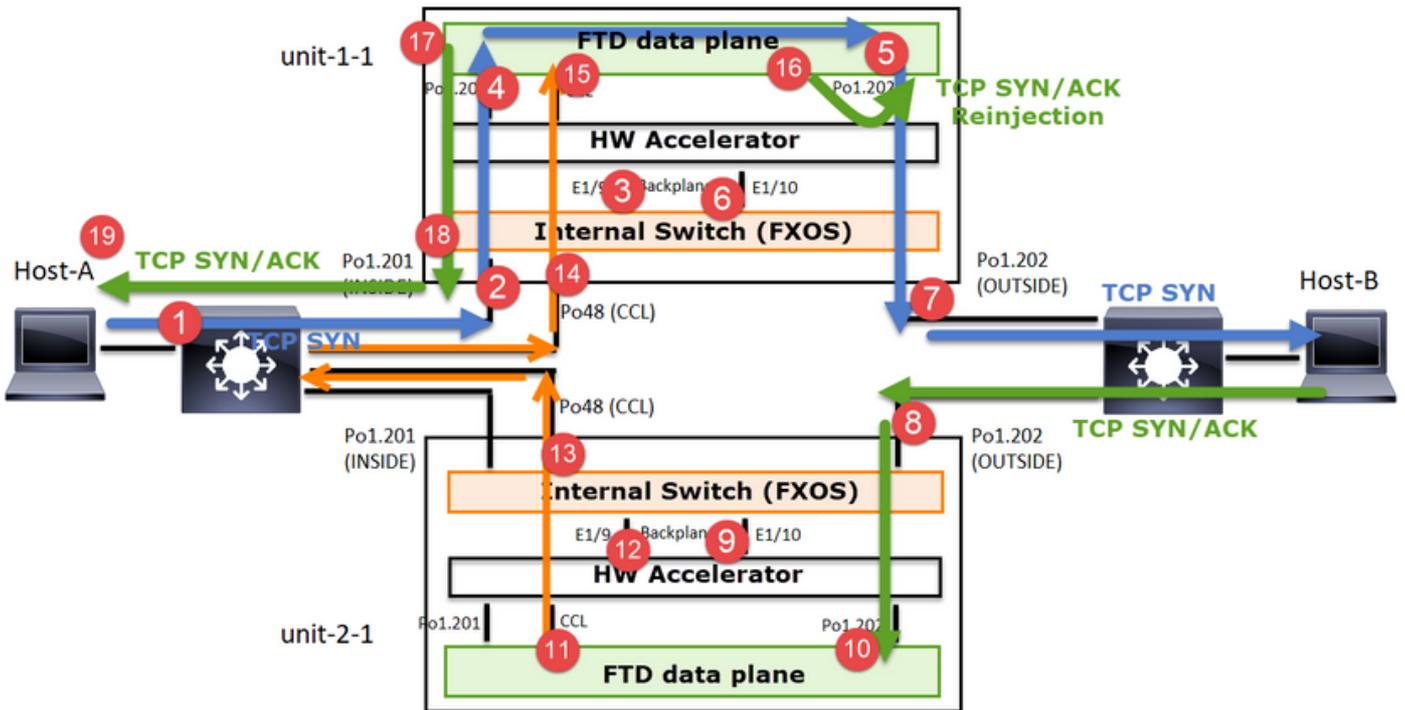
Firepower 어플라이언스는 전송 흐름에 대한 가시성을 제공하는 여러 캡처 지점을 제공합니다. 문제를 해결하고 클러스터를 활성화하면 다음과 같은 주요 문제가 발생합니다.

- 캡처 수는 클러스터의 유닛 수가 증가함에 따라 증가합니다.
- 클러스터가 특정 플로우를 처리하는 방법을 알아야 클러스터를 통해 패킷을 추적할 수 있습니다.

이 다이어그램은 2유닛 클러스터(예: FP941xx/FP9300)를 보여줍니다.



비대칭 TCP 연결 설정의 경우 TCP SYN, SYN/ACK 교환은 다음과 같습니다.



### 전달 트래픽

1. TCP SYN은 Host-A에서 Host-B로 전송됩니다.
2. TCP SYN이 새시(Po1 멤버 중 하나)에 도착합니다.
3. TCP SYN은 새시 백플레인 인터페이스(예: E1/9, E1/10 등) 중 하나를 통해 데이터 플레인으로 전송됩니다.
4. TCP SYN은 데이터 플레인 인그레스 인터페이스(Po1.201/INSIDE)에 도착합니다. 이 예에서 unit1-1은 플로우의 소유권을 가져오고, ISN(Initial Sequence Number) 임의 지정을 수행하고, 소유권(쿠키) 정보를 Seq 번호로 인코딩합니다.

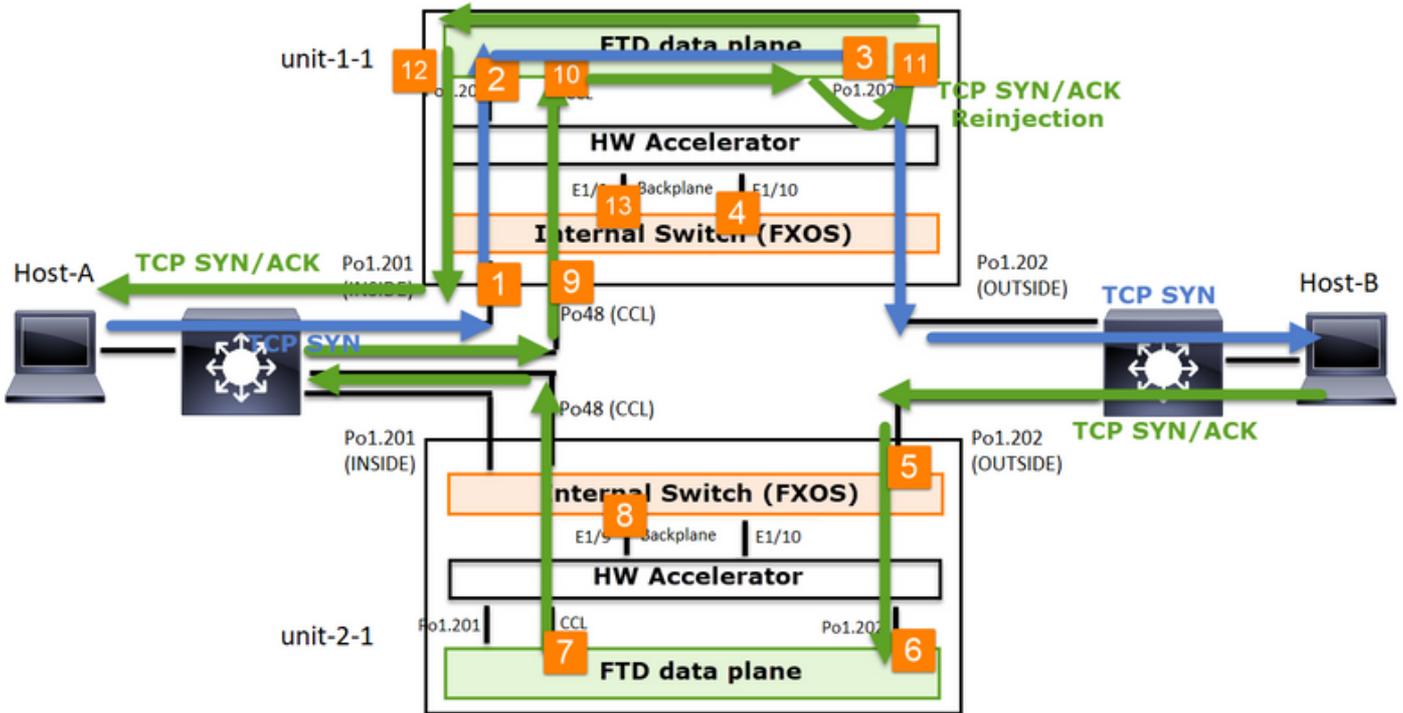
5. TCP SYN은 Po1.202/OUTSIDE(데이터 플레인 이그레스 인터페이스)에서 전송됩니다.
6. TCP SYN은 새시 백플레인 인터페이스(예: E1/9, E1/10 등) 중 하나에 도착합니다.
7. TCP SYN은 새시 물리적 인터페이스(Po1 멤버 중 하나)에서 Host-B로 전송됩니다.

#### 반환 트래픽

8. TCP SYN/ACK가 Host-B에서 전송되어 unit-2-1(Po1의 멤버 중 하나)에 도착합니다.
9. TCP SYN/ACK는 새시 백플레인 인터페이스 중 하나(예: E1/9, E1/10 등)를 통해 데이터 플레인으로 전송됩니다.
10. TCP SYN/ACK가 데이터 플레인 이그레스 인터페이스(Po1.202/OUTSIDE)에 도착합니다.
11. TCP SYN/ACK가 CCL(Cluster Control Link)에서 유닛-1-1로 전송됩니다. 기본적으로 ISN이 활성화되어 있습니다. 따라서 전달자는 디렉터의 개입 없이 TCP SYN+ACK에 대한 소유자 정보를 찾습니다. 다른 패킷의 경우 또는 ISN이 비활성화된 경우 디렉터를 쿼리합니다.
12. TCP SYN/ACK는 새시 백플레인 인터페이스(예: E1/9, E1/10 등) 중 하나에 도착합니다.
13. TCP SYN/ACK는 새시 물리적 인터페이스(Po48 멤버 중 하나)에서 unit-1-1로 전송됩니다.
14. TCP SYN/ACK가 unit-1-1(Po48 멤버 중 하나)에 도착합니다.
15. TCP SYN/ACK는 새시 백플레인 인터페이스 중 하나를 통해 데이터 플레인 CCL 포트 채널 인터페이스(nameif 클러스터)로 전달됩니다.
16. 데이터 플레인에서는 TCP SYN/ACK 패킷을 데이터 플레인 인터페이스 Po1.202/OUTSIDE로 다시 전달합니다.
17. TCP SYN/ACK는 Po1.201/INSIDE(데이터 플레인 이그레스 인터페이스)에서 HOST-A로 전송됩니다.
18. TCP SYN/ACK는 새시 백플레인 인터페이스 중 하나(예: E1/9, E1/10 등)를 통과하고 Po1의 멤버 중 하나를 이그레스(egress)합니다.
19. TCP SYN/ACK가 호스트 A에 도착합니다.

이 시나리오에 대한 자세한 내용은 Cluster Connection Establishment Case Studies(클러스터 연결 설정 사례 연구)의 관련 섹션을 참조하십시오.

이 패킷 교환을 기반으로 가능한 모든 클러스터 캡처 포인트는 다음과 같습니다.



### 전달 트래픽(예: TCP SYN) 캡처의 경우

1. 새시 물리적 인터페이스(예: Po1 멤버). 이 캡처는 CM(Chassis Manager) UI 또는 CM CLI에서 구성합니다.
2. 데이터 플레인 인그레스 인터페이스(예: Po1.201 INSIDE).
3. 데이터 플레인 이그레스 인터페이스(예: Po1.202 OUTSIDE).
4. 새시 백플레인 인터페이스. FP4100에는 2개의 백플레인 인터페이스가 있습니다. FP9300에는 총 6개(모듈당 2개)가 있습니다. 패킷이 어느 인터페이스에 도착하는지 알 수 없으므로 모든 인터페이스에서 캡처를 활성화해야 합니다.

### 반환 트래픽(예: TCP SYN/ACK) 캡처의 경우

5. 새시 물리적 인터페이스(예: Po1 멤버). 이 캡처는 CM(Chassis Manager) UI 또는 CM CLI에서 구성합니다.
6. 데이터 평면 인그레스 인터페이스(예: Po1.202 OUTSIDE).
7. 패킷이 리디렉션되므로 다음 캡처 포인트는 데이터 플레인 CCL입니다.
8. 새시 백플레인 인터페이스. 두 인터페이스에서 모두 캡처를 활성화해야 합니다.
9. Unit-1-1 새시 CCL 멤버 인터페이스.
10. 데이터 플레인 CCL 인터페이스(nameif 클러스터).
11. 인그레스 인터페이스(Po1.202 OUTSIDE) CCL에서 데이터 평면으로 재삽입된 패킷입니다.
12. 데이터 플레인 이그레스 인터페이스(예: Po1.201 INSIDE).
13. 새시 백플레인 인터페이스.

### 클러스터 캡처를 활성화하는 방법

#### FXOS 캡처

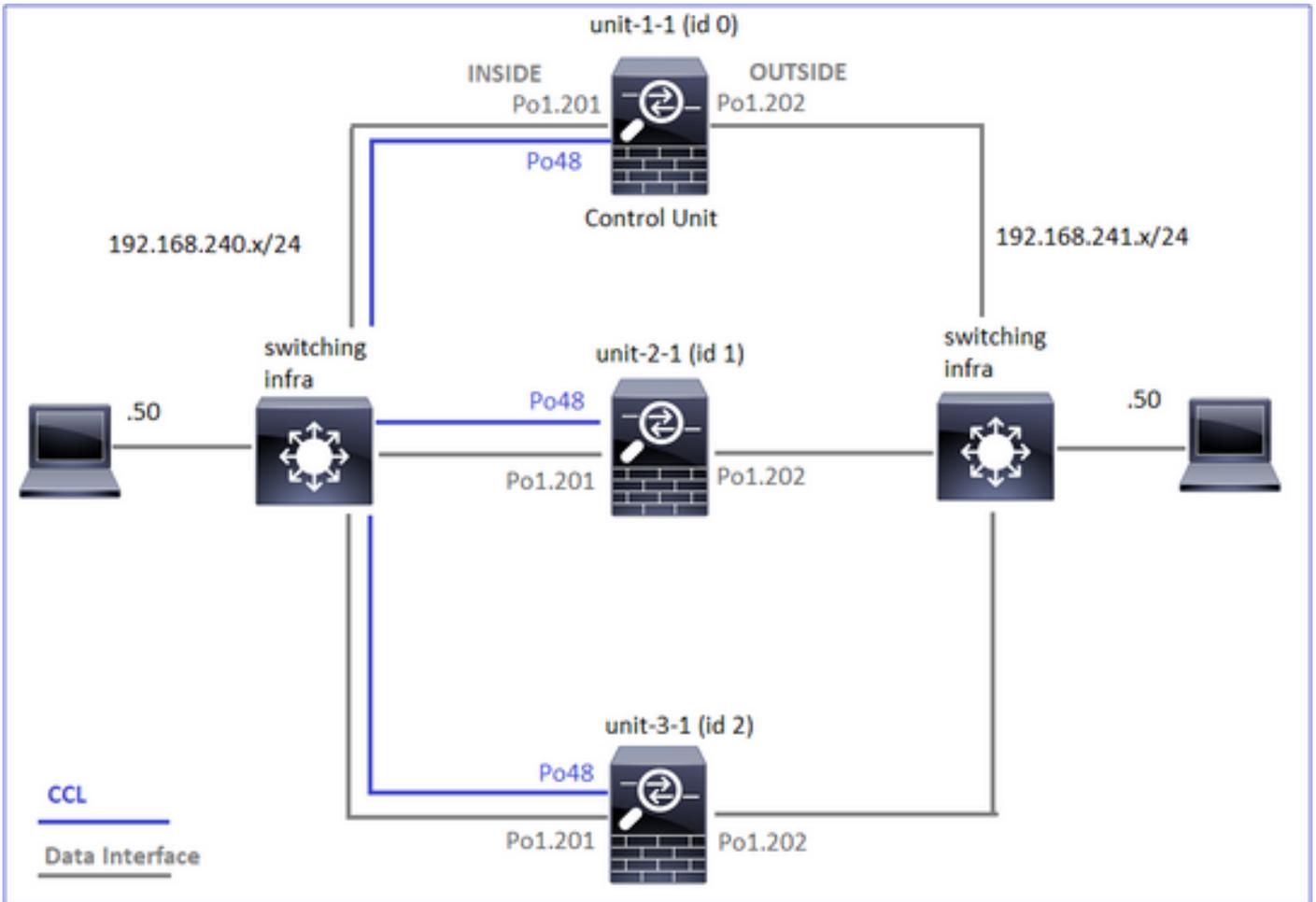
이 프로세스는 FXOS 컨피그레이션 가이드에서 설명합니다. [패킷 캡처](#)

 참고: FXOS 캡처는 내부 스위치 관점에서 인그레스 방향으로만 수행할 수 있습니다.

## 데이터 플레인 캡처

모든 클러스터 멤버에서 캡처를 활성화하는 권장 방법은 cluster exec 명령을 사용하는 것입니다.

3-유닛 클러스터를 고려합니다.



모든 클러스터 유닛에 활성 캡처가 있는지 확인하려면 다음 명령을 사용합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show capture
```

```
unit-1-1(LOCAL):*****
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

```
firepower#
```

Po1.201(INSIDE)의 모든 유닛에서 데이터 플레인 캡처를 활성화하려면 다음을 수행합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec capture CAPI interface INSIDE
```

캡처 필터를 지정하고, 트래픽이 많을 것으로 예상되는 경우 캡처 버퍼를 늘리는 것이 좋습니다.

```
<#root>
```

```
firepower#
```

```
cluster exec capture CAPI buffer 33554432 interface INSIDE match tcp host 192.168.240.50 host 192.168.24
```

확인

```
<#root>
```

```
firepower#
```

```
cluster exec show capture
```

```
unit-1-1(LOCAL):*****  
capture CAPI type raw-data buffer 33554432 interface INSIDE [Capturing - 5140 bytes]  
  match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-2-1:*****  
capture CAPI type raw-data buffer 33554432 interface INSIDE [Capturing - 260 bytes]  
  match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-3-1:*****  
capture CAPI type raw-data buffer 33554432 interface INSIDE [Capturing - 0 bytes]  
  match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

모든 캡처의 내용을 보려면(이 출력은 매우 길 수 있음)

```
<#root>
```

```
firepower#
```

```
terminal pager 24
```

```
firepower#
```

```
cluster exec show capture CAPI
```

unit-1-1(LOCAL):\*\*\*\*\*

21 packets captured

```
1: 11:33:09.879226 802.1Q vlan#201 PO 192.168.240.50.45456 > 192.168.241.50.80: S 2225395909:2225395909
2: 11:33:09.880401 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.45456: S 719653963:719653963(0
3: 11:33:09.880691 802.1Q vlan#201 PO 192.168.240.50.45456 > 192.168.241.50.80: . ack 719653964 win 229
4: 11:33:09.880783 802.1Q vlan#201 PO 192.168.240.50.45456 > 192.168.241.50.80: P 2225395910:2225396054
```

unit-2-1:\*\*\*\*\*

0 packet captured

0 packet shown

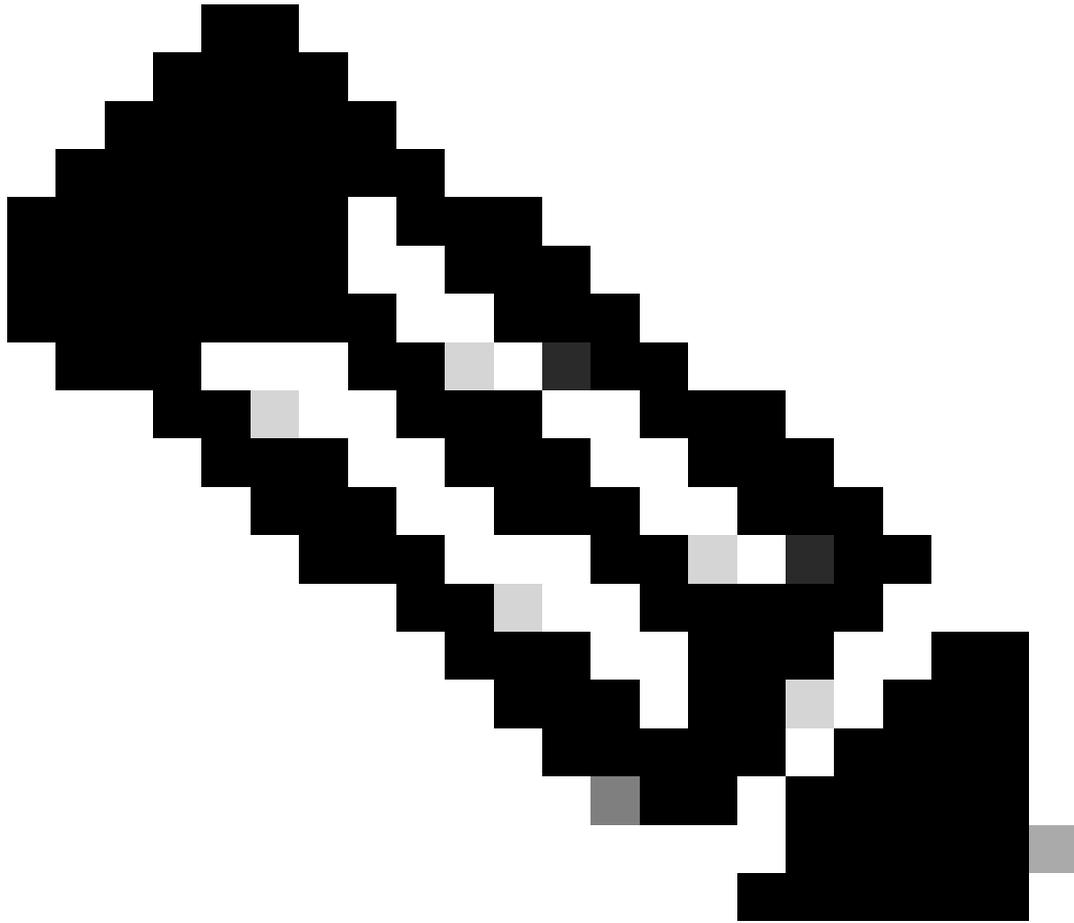
unit-3-1:\*\*\*\*\*

0 packet captured

0 packet shown

## 추적 캡처

각 유닛의 데이터 플레인에서 인그레스 패킷을 처리하는 방법을 보려면 trace 키워드를 사용합니다. 그러면 처음 50개의 인그레스 패킷이 추적됩니다. 최대 1000개의 인그레스 패킷을 추적할 수 있습니다.



참고: 인터페이스에 여러 캡처가 적용된 경우 단일 패킷을 한 번만 추적할 수 있습니다.

모든 클러스터 유닛의 인터페이스 OUTSIDE에서 첫 1,000개의 인그레스 패킷을 추적하려면

```
<#root>
```

```
firepower#
```

```
cluster exec cap CAPO int OUTSIDE buff 33554432 trace trace-count 1000 match tcp host 192.168.240.50 hos
```

관심 플로우를 캡처한 후에는 각 유닛의 관심 패킷을 추적해야 합니다. 기억해야 할 중요한 것은 특정 패킷을 유닛-1-1에 #1 수 있지만 다른 유닛에 #2 수 있다는 것입니다.

이 예에서는 SYN/ACK가 unit-2-1에서는 패킷 #2, unit-3-1에서는 패킷 #1을 확인할 수 있습니다.

```
<#root>
```

firepower#

cluster exec show capture CAPO | include S.\*ack

```
unit-1-1(LOCAL):*****
1: 12:58:31.117700 802.1Q vlan#202 PO 192.168.240.50.45468 > 192.168.241.50.80: S 441626016:441626016(0)
2: 12:58:31.118341 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:
s
301658077:301658077(0)
```

```
ack
441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
```

unit-2-1:\*\*\*\*\*

```
unit-3-1:*****
1: 12:58:31.111429 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:
s
301658077:301658077(0)
```

```
ack
441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
```

로컬 유닛에서 패킷 #2(SYN/ACK)를 추적하려면 다음을 수행합니다.

<#root>

firepower#

cluster exec show cap CAPO packet-number 2 trace

```
unit-1-1(LOCAL):*****
2: 12:58:31.118341 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:
s
301658077:301658077(0)
```

```
ack
441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
MAC Access list
...
```

원격 유닛에서 동일한 패킷(SYN/ACK)을 추적하려면

<#root>

firepower#

```
cluster exec unit unit-3-1 show cap CAPO packet-number 1 trace
```

```
1: 12:58:31.111429 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45468:
```

s

```
301658077:301658077(0)
```

ack

```
441626017 win 28960 <mss 1460,sackOK,timestamp 1125686319 1115330849,nop,wscale 7>
```

Phase: 1

Type: CAPTURE

Subtype:

Result: ALLOW

Config:

Additional Information:

MAC Access list

...

## CCL 캡처

CCL 링크(모든 장치에서)에서 캡처를 활성화하려면

<#root>

firepower#

```
cluster exec capture CCL interface cluster
```

```
unit-1-1(LOCAL):*****
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

## 다시 삽입 숨기기

기본적으로 데이터 플레인 데이터 인터페이스에서 활성화된 캡처는 모든 패킷을 표시합니다.

- 물리적 네트워크에서 전송되는 것은
- CCL에서 재삽입된 것

재삽입된 패킷을 표시하지 않으려면 `reinject-hide` 옵션을 사용합니다. 이는 플로우가 비대칭적인지 확인하려는 경우 유용할 수 있습니다.

<#root>

```
firepower#
```

```
cluster exec capture CAPI_RH reinject-hide interface INSIDE match tcp host 192.168.240.50 host 192.168.2
```

이 캡처는 로컬 유닛이 실제로 특정 인터페이스에서 다른 클러스터 유닛이 아닌 물리적 네트워크로 부터 직접 수신한 것을 보여줍니다.

### ASP 삭제

특정 흐름에 대한 소프트웨어 삭제를 확인하려면 asp-drop capture를 활성화할 수 있습니다. 초점을 두어야 할 삭제 이유를 모를 경우 all 키워드를 사용합니다. 또한 패킷 페이로드에 관심이 없는 경우 headers-only 키워드를 지정할 수 있습니다. 이렇게 하면 20-30배 많은 패킷을 캡처할 수 있습니다.

```
<#root>
```

```
firepower#
```

```
cluster exec cap ASP type asp-drop all buffer 33554432 headers-only
```

```
unit-1-1(LOCAL):*****
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

또한 ASP 캡처에서 원하는 IP를 지정할 수 있습니다.

```
<#root>
```

```
firepower#
```

```
cluster exec cap ASP type asp-drop all buffer 33554432 headers-only
```

```
match ip host 192.0.2.100 any
```

### 캡처 지우기

모든 클러스터 단위로 실행되는 모든 캡처의 버퍼를 지웁니다. 이렇게 하면 캡처가 중지되지 않고 버퍼만 지워집니다.

```
<#root>
```

```
firepower#
```

```
cluster exec clear capture /all
```

```
unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

## 캡처 중지

모든 클러스터 유닛에서 활성 캡처를 중지하는 방법에는 두 가지가 있습니다. 나중에 다시 시작할 수 있습니다.

### 방법 1

```
<#root>
firepower#
cluster exec cap CAPI stop

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

### 다시 시작하려면

```
<#root>
firepower#
cluster exec no capture CAPI stop

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

### 2단계

```
<#root>
firepower#
cluster exec no capture CAPI interface INSIDE
```

```
unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

다시 시작하려면

<#root>

firepower#

cluster exec capture CAPI interface INSIDE

```
unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

### 캡처 수집

캡처를 내보내는 방법에는 여러 가지가 있습니다.

#### 방법 1 - 원격 서버로

이렇게 하면 데이터 플레인에서 원격 서버(예: TFTP)로 캡처를 업로드할 수 있습니다. 소스 유닛을 반영하도록 캡처 이름이 자동으로 변경됩니다.

<#root>

firepower#

cluster exec copy /pcap capture:CAPI tftp://192.168.240.55/CAPI.pcap

```
unit-1-1(LOCAL):*****
```

Source capture name [CAPI]?

Address or name of remote host [192.168.240.55]?

Destination filename [CAPI.pcap]?

INFO: Destination filename is changed to unit-1-1\_CAPI.pcap !!!!!!!

81 packets copied in 0.40 secs

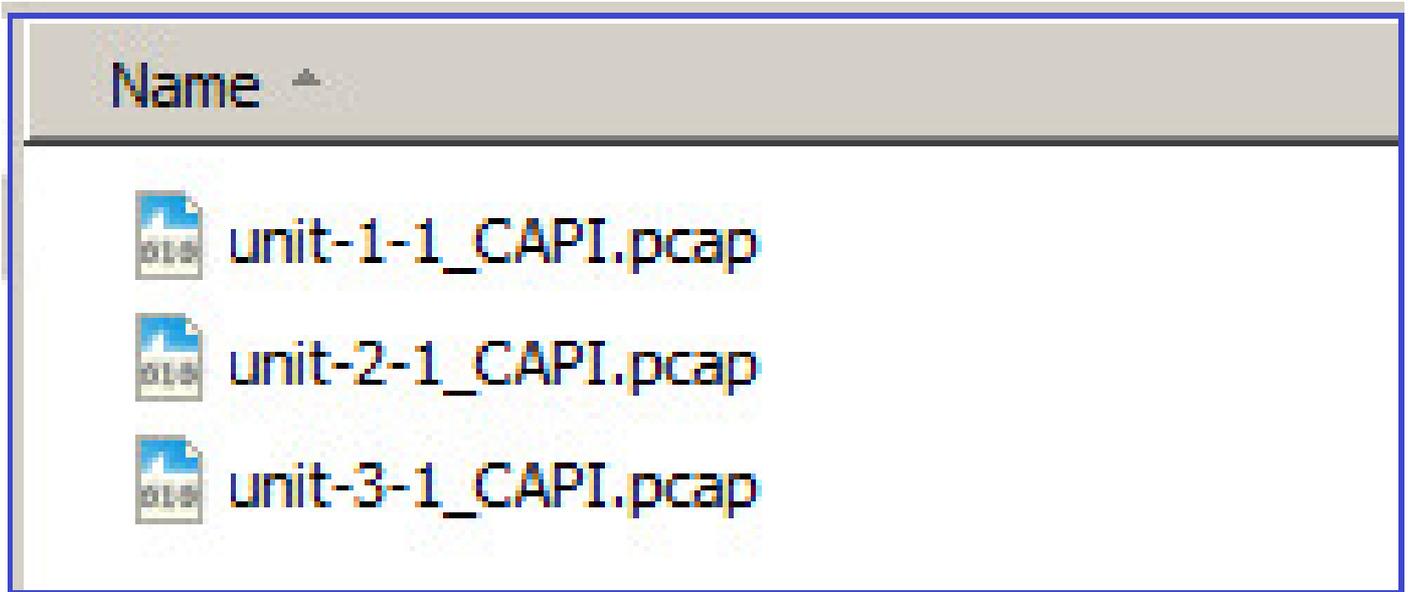
```
unit-2-1:*****
```

INFO: Destination filename is changed to unit-2-1\_CAPI.pcap !

unit-3-1:\*\*\*\*\*

INFO: Destination filename is changed to unit-3-1\_CAPI.pcap !

업로드된 pcap 파일:



Way 2 - FMC에서 캡처 가져오기

이 방법은 FTD에만 적용됩니다. 먼저 캡처를 FTD 디스크에 복사합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec copy /pcap capture:CAPI disk0:CAPI.pcap
```

unit-1-1(LOCAL):\*\*\*\*\*

```
Source capture name [CAPI]?
```

```
Destination filename [CAPI.pcap]?
```

```
!!!!
```

```
62 packets copied in 0.0 secs
```

expert 모드에서 파일을 /mnt/disk0/에서 /ngfw/var/common/ 디렉토리로 복사합니다.

```
<#root>
```

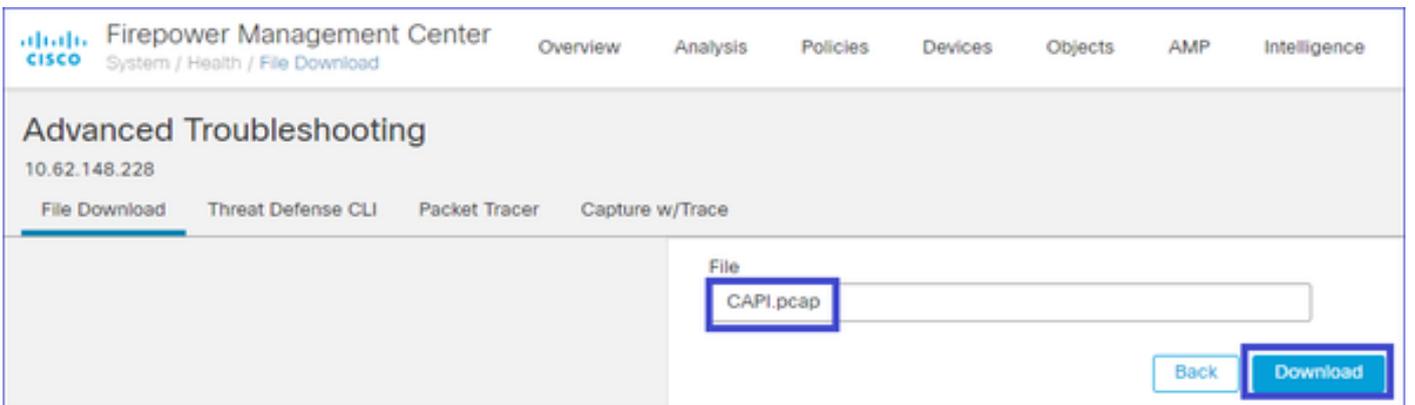
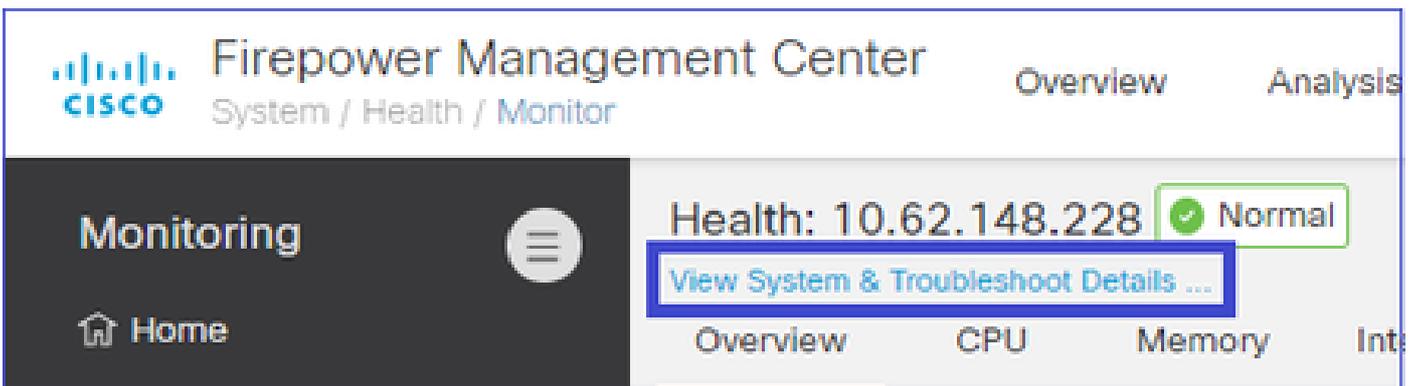
```
>
```

```
expert
```

```
admin@firepower:~$
cd /mnt/disk0

admin@firepower:/mnt/disk0$
sudo cp CAPI.pcap /ngfw/var/common
```

마지막으로 FMC에서 System > Health > Monitor 섹션으로 이동합니다. View System & Troubleshoot Details(시스템 및 문제 해결 세부사항 보기) > Advanced Troubleshooting(고급 문제 해결)을 선택하고 캡처 파일을 가져옵니다.



캡처 삭제

모든 클러스터 유닛에서 캡처를 제거하려면 다음 명령을 사용합니다.

```
<#root>
firepower#
cluster exec no capture CAPI

unit-1-1(LOCAL):*****
unit-2-1:*****
unit-3-1:*****
```

## 오프로드된 플로우

FP41xx/FP9300에서 플로우는 정적(예: Fastpath 규칙) 또는 동적으로 HW Accelerator로 오프로드 될 수 있습니다. 플로우 오프로드에 대한 자세한 내용은 다음 문서를 참조하십시오.

<https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/212321-clarify-the-firepower-threat-defense-acc.html#anc22>

플로우가 오프로드되면 일부 패킷만 FTD 데이터 플레인을 통과합니다. 나머지는 HW 가속기 (Smart NIC)에서 처리합니다.

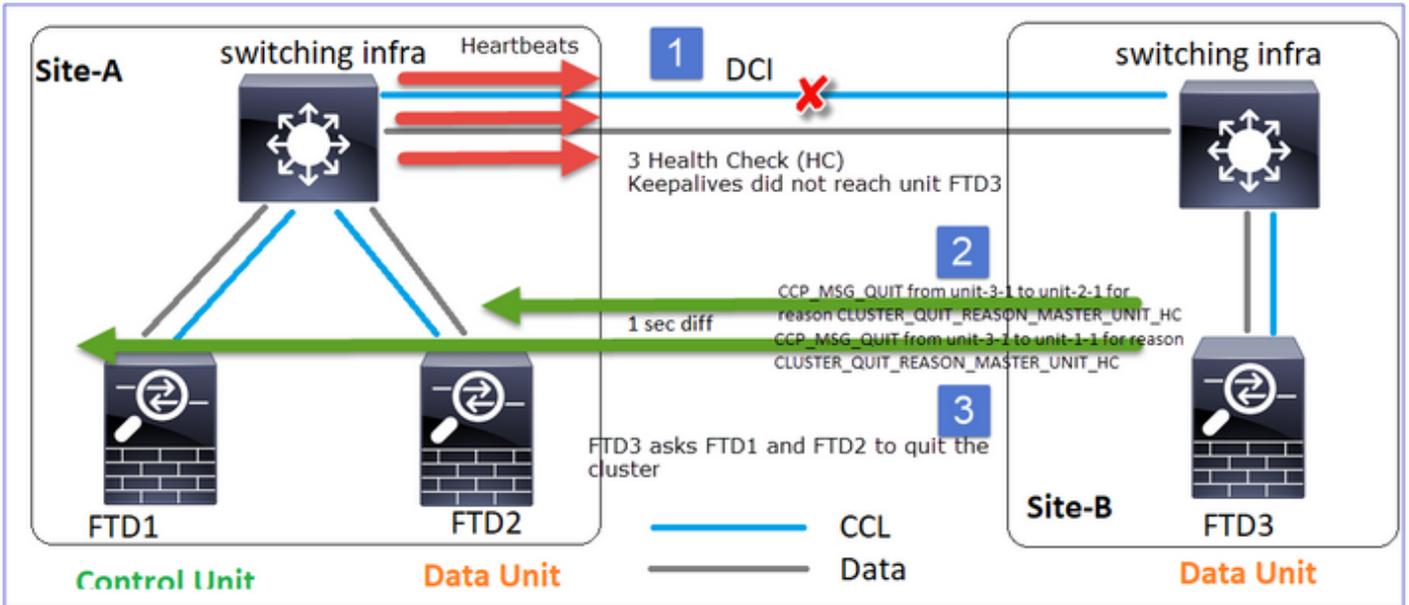
캡처 관점에서 보면, 이는 FTD 데이터 플레인 레벨 캡처만 활성화할 경우 디바이스를 통과하는 모든 패킷이 표시되지 않음을 의미합니다. 이 경우 FXOS 새시 레벨 캡처도 활성화해야 합니다.

## CCL(Cluster Control Link) 메시지

CCL에서 캡처를 수행하면 클러스터 유닛에서 서로 다른 유형의 메시지를 교환하는 것을 확인할 수 있습니다. 관심 분야:

프로토콜	설명
UDP 49495	<p>클러스터 하트비트(킵얼라이브)</p> <ul style="list-style-type: none"> <li>· L3 브로드캐스트(255.255.255.255)</li> <li>· 이러한 패킷은 모든 클러스터 유닛에서 상태 점검 보류 시간 값의 1/3로 전송됩니다.</li> <li>· 캡처에 표시된 모든 UDP 49495 패킷이 하트비트는 아닙니다</li> <li>· 하트비트에는 시퀀스 번호가 포함됩니다.</li> </ul>
UDP 4193	<p>클러스터 제어 프로토콜 데이터 경로 메시지</p> <ul style="list-style-type: none"> <li>· 유니캐스트</li> <li>· 이러한 패킷에는 플로우 소유자, 디렉터, 백업 소유자 등에 대한 정보(메타 데이터)가 포함됩니다. 예: <ul style="list-style-type: none"> <li>· 새 플로우가 생성되면 소유자로부터 디렉터로 '클러스터 추가' 메시지가 전송됩니다.</li> <li>· 플로우가 종료되면 소유자로부터 디렉터로 '클러스터 삭제' 메시지가 전송됩니다.</li> </ul> </li> </ul>
데이터 패킷	클러스터를 통과하는 다양한 트래픽 흐름에 속하는 데이터 패킷



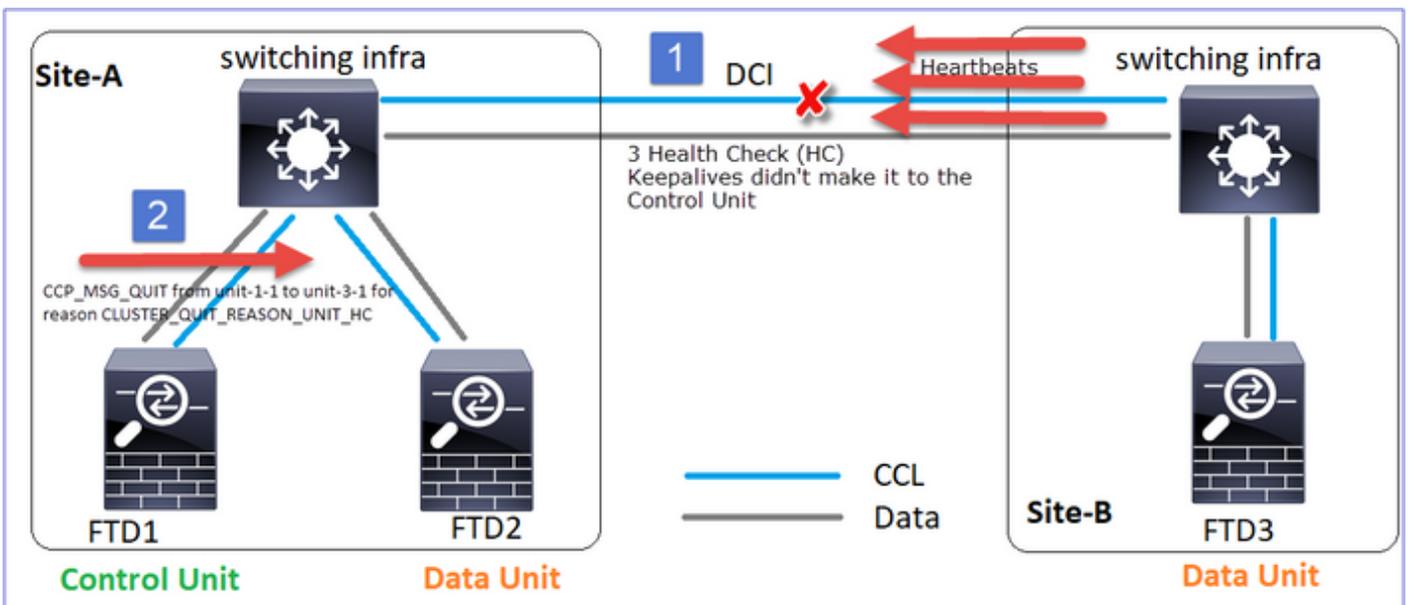


Q. CLUSTER\_QUIT\_REASON\_PRIMARY\_UNIT\_HC의 목적은 무엇입니까?

A. unit-3-1(Site-B)의 관점에서 볼 때 사이트 A에서 unit-1-1 및 unit-2-1에 대한 연결이 모두 끊기므로 가능한 한 빨리 구성원 목록에서 제거해야 합니다. 그렇지 않으면 unit-2-1이 여전히 구성원 목록에 있고 unit-2-1이 연결의 디렉터인 경우 패킷이 손실될 수 있으며 unit-2-1에 대한 흐름 쿼리가 실패합니다.

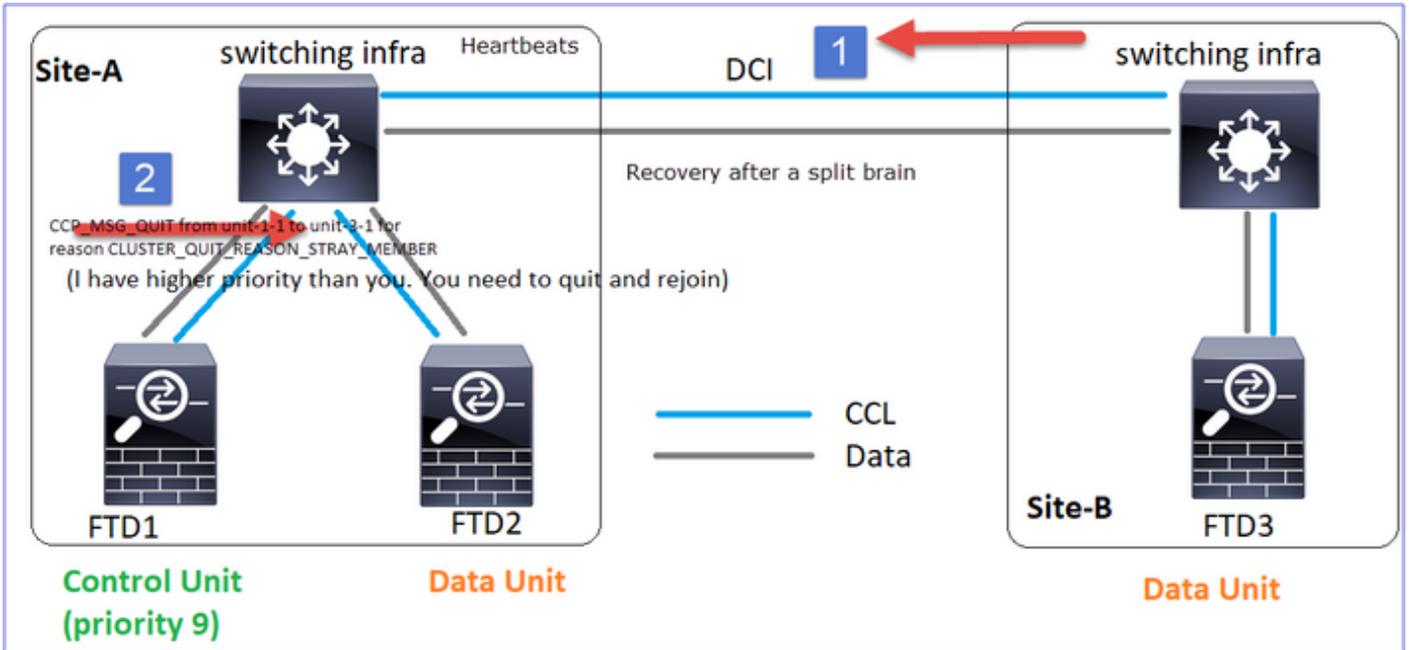
#### CLUSTER\_QUIT\_REASON\_UNIT\_HC

제어 노드는 데이터 노드에서 3개의 연속 하트비트 메시지를 잃을 때마다 CCL을 통해 CLUSTER\_QUIT\_REASON\_UNIT\_HC 메시지를 전송합니다. 이 메시지는 유니캐스트입니다.



#### CLUSTER\_QUIT\_REASON\_STRAY\_MEMBER

분할 파티션이 피어 파티션과 다시 연결되면, 새로운 데이터 노드는 우세 제어 유닛에 의해 미처 멤버로 취급되며 CLUSTER\_QUIT\_REASON\_STRAY\_MEMBER를 이유로 하는 CCP 종료 메시지를 받습니다.



### CLUSTER\_QUIT\_MEMBER\_DROPOUT

데이터 노드에 의해 생성되고 브로드캐스트로 전송되는 브로드캐스트 메시지입니다. 유닛에서 이 메시지를 수신하면 DISABLED 상태로 이동합니다. 또한 자동 재연결은 킥오프되지 않습니다.

```
<#root>
```

```
firepower#
```

```
show cluster info trace | include DROPOUT
```

```
Nov 04 00:22:54.699 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-1-1 for reason
CLUSTER_QUIT_MEMBER_DROPOUT
```

```
Nov 04 00:22:53.699 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-2-1 for reason
CLUSTER_QUIT_MEMBER_DROPOUT
```

클러스터 내역은 다음과 같습니다.

```
<#root>
```

```
PRIMARY      DISABLED      Received control message DISABLE (
member dropout announcement
)
```

### HC(Cluster Health-Check) 메커니즘

## 중요 사항

- 각 클러스터 유닛은 상태 확인 보류 시간 값의 1/3마다 하트비트를 다른 모든 유닛(브로드캐스트 255.255.255.255)으로 전송하고 CCL을 통한 전송으로 UDP 포트 49495을 사용합니다.
- 각 클러스터 유닛은 Poll 타이머와 Poll 카운트 값을 사용하여 다른 모든 유닛을 독립적으로 추적합니다.
- 클러스터 유닛이 하트비트 간격 내에 클러스터 피어 유닛으로부터 어떤 패킷(하트비트 또는 데이터 패킷)도 수신하지 못하면 Poll count 값이 증가합니다.
- 클러스터 피어 유닛의 폴링 카운트 값이 3이 되면 피어는 중단된 것으로 간주됩니다.
- 하트비트가 수신될 때마다, 그 시퀀스 번호가 체크되고 이전에 수신된 하트비트와의 차이가 1과 다른 경우, 하트비트 드롭 카운터가 그에 따라 증가한다.
- 클러스터 피어에 대한 Poll 카운트 카운터가 0과 다르고 피어에서 패킷을 수신하면 카운터가 0으로 재설정됩니다.

클러스터 상태 카운터를 확인하려면 다음 명령을 사용합니다.

```
<#root>
```

```
firepower#
```

```
show cluster info health details
```

Unit (ID)	Heartbeat count	Heartbeat drops	Average gap (ms)	Maximum slip (ms)	Poll count
unit-2-1 ( 1)	650	0	4999	1	0
unit-3-1 ( 2)	650	0	4999	1	0

## 기본 열에 대한 설명

열	설명
장치(ID)	원격 클러스터 피어의 ID입니다.
하트비트 수	CCL을 통해 원격 피어에서 받은 하트비트 수입니다.
하트비트 삭제	누락된 하트비트 수입니다. 이 카운터는 수신된 하트비트 시퀀스 번호에 기초하여 계산된다.
평균 격차	수신된 하트비트의 평균 시간 간격입니다.

폴링 카운트

이 카운터가 30이 되면 유닛이 클러스터에서 제거됩니다. 폴링  
쿼리 간격은 하트비트 간격과 동일하지만 독립적으로 실행됩  
니다.

카운터를 재설정하려면 다음 명령을 사용합니다.

<#root>

firepower#

```
clear cluster info health details
```

Q. 하트비트 빈도를 어떻게 확인합니까?

A. 평균 간격 값을 확인합니다.

<#root>

firepower#

```
show cluster info health details
```

```
-----  
|                Unit (ID)| Heartbeat| Heartbeat|
```

Average

```
| Maximum|      Poll|  
|                | count|      drops|
```

gap (ms)

```
| slip (ms)|      count|
```

```
-----  
|                unit-2-1 ( 1)|      3036|           0|
```

999

```
|           1|           0|
```

Q. FTD에서 클러스터 보류 시간을 변경하려면 어떻게 해야 합니까?

A. FlexConfig 사용

Q. 스플릿 브레인 후 누가 제어 노드가 됩니까?

A. 우선순위가 가장 높은(가장 낮은) 장치:

<#root>

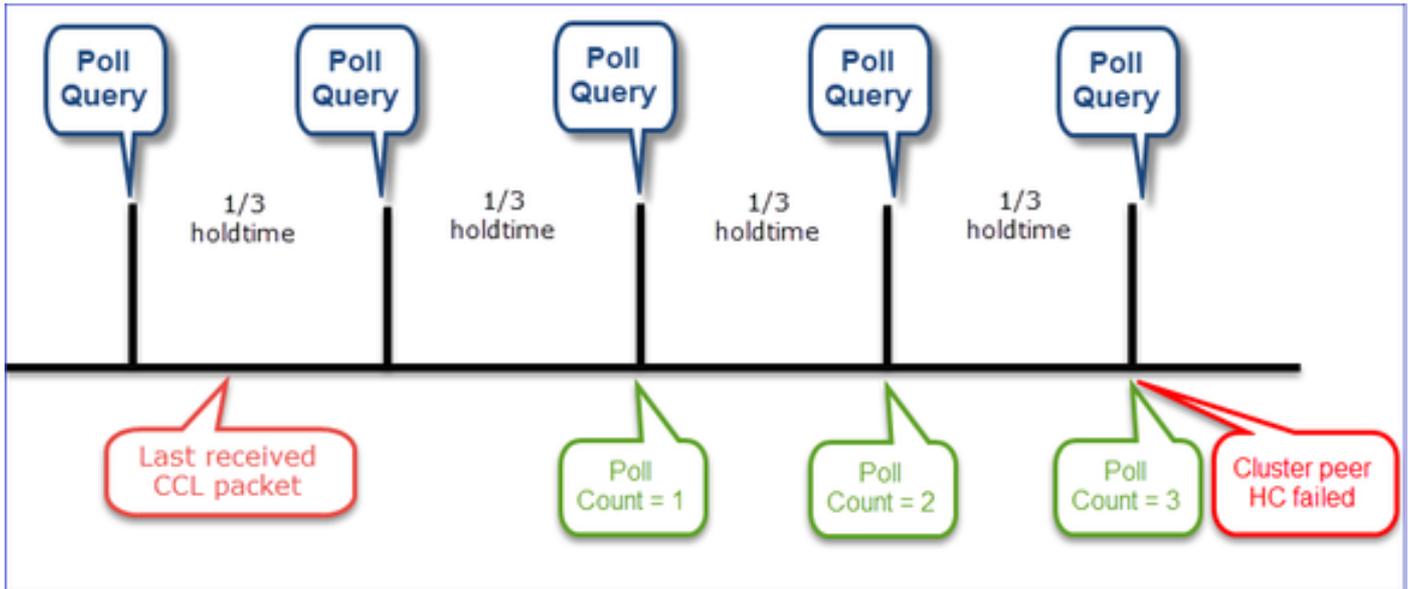
firepower#

show run cluster | include priority

priority 9

자세한 내용은 HC 실패 시나리오 1을 참조하십시오.

클러스터 HC 메커니즘 시각화



알림 타이머: 최소값과 최대값은 마지막으로 수신된 CCL 패킷 도착에 따라 달라집니다.

보류 시간	쿼리 검사 폴링 (빈도)	최소 탐지 시간	최대 탐지 시간
3초(기본값)	~1초	~3.01초	~3.99초
4초	~1.33초	~4.01초	~5.32초
5초	~1.66초	~5.01초	~6.65초
6초	~2초	~6.01초	~7.99초
7초	~2.33초	~7.01초	~9.32초

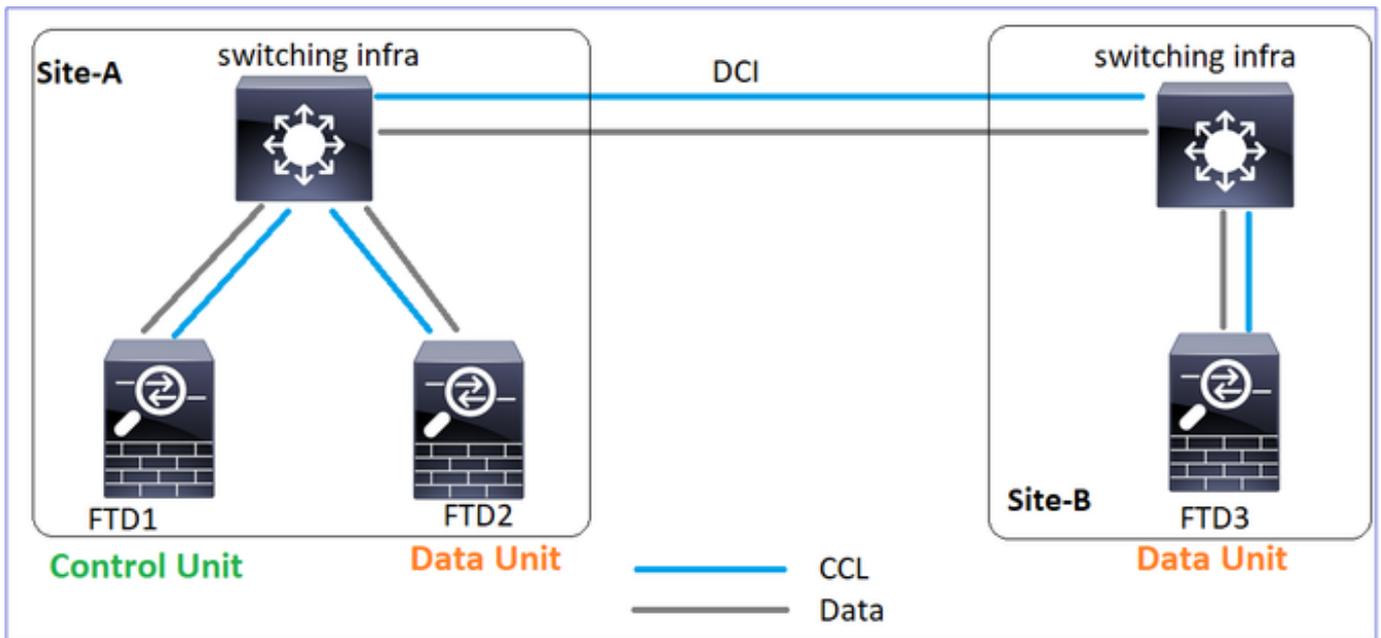
8초	~2.66초	~8.01초	~10.65초
----	--------	--------	---------

## 클러스터 HC 오류 시나리오

이 섹션의 목표는 다음을 시연하는 것입니다.

- 다른 클러스터 HC 실패 시나리오
- 서로 다른 로그와 명령 출력의 상관 관계 분석 방법

토폴로지



## 클러스터 컨피그레이션

Unit-1-1	Unit-2-1
<pre> cluster group GROUP1 key ***** local-unit unit-1-1 cluster-interface Port-channel48 ip 10.17.1.1 255.255.0.0 priority 9 health-check holdtime 3 health-check data-interface auto-rejoin 3 5 2 health-check cluster-interface auto-rejoin unlimited 5 1 health-check system auto-rejoin 3 5 2 health-check monitor-interface debounce-time 500 site-id 1 enable </pre>	<pre> cluster group GROUP1 key ***** local-unit unit-2-1 cluster-interface Port-channel48 ip 10.17.1.2 255.255.0.0 priority 17 health-check holdtime 3 health-check data-interface auto-rejoin 3 5 2 health-check cluster-interface auto-rejoin unlimited 5 1 health-check system auto-rejoin 3 5 2 health-check monitor-interface debounce-time 500 site-id 1 enable </pre>

클러스터 상태

Unit-1-1	Unit-2-1
<pre>&lt;#root&gt; firepower# show cluster info  Cluster GROUP1: On   Interface mode: spanned  This is "unit-1-1" in state PRIMARY        ID      : 0       Site ID  : 1       Version  : 9.12(2)33       Serial No.: FCH22247LNK       CCL IP   : 10.17.1.1       CCL MAC  : 0015.c500.018f       Last join : 20:25:36 UTC Nov 1 2020       Last leave: 20:25:28 UTC Nov 1 2020 Other members in the cluster:  Unit "unit-3-1" in state secondary        ID      : 1       Site ID  : 2       Version  : 9.12(2)33       Serial No.: FCH22247MKJ       CCL IP   : 10.17.3.1       CCL MAC  : 0015.c500.038f       Last join : 20:58:45 UTC Nov 1 2020       Last leave: 20:58:37 UTC Nov 1 2020  Unit "unit-2-1" in state SECONDARY        ID      : 2       Site ID  : 1       Version  : 9.12(2)33       Serial No.: FCH23157Y9N       CCL IP   : 10.17.2.1       CCL MAC  : 0015.c500.028f       Last join : 20:44:45 UTC Nov 1 2020       Last leave: 20:44:38 UTC Nov 1 2020</pre>	<pre>&lt;#root&gt; firepower# show cluster info  Cluster GROUP1: On   Interface mode: spanned  This is "unit-2-1" in state SECONDARY        ID      : 2       Site ID  : 1       Version  : 9.12(2)33       Serial No.: FCH23157Y9N       CCL IP   : 10.17.2.1       CCL MAC  : 0015.c500.028f       Last join : 20:44:46 UTC Nov 1 2020       Last leave: 20:44:38 UTC Nov 1 2020 Other members in the cluster:  Unit "unit-1-1" in state PRIMARY        ID      : 0       Site ID  : 1       Version  : 9.12(2)33       Serial No.: FCH22247LNK       CCL IP   : 10.17.1.1       CCL MAC  : 0015.c500.018f       Last join : 20:25:36 UTC Nov 1 2020       Last leave: 20:25:28 UTC Nov 1 2020  Unit "unit-3-1" in state SECONDARY        ID      : 1       Site ID  : 2       Version  : 9.12(2)33       Serial No.: FCH22247MKJ       CCL IP   : 10.17.3.1       CCL MAC  : 0015.c500.038f       Last join : 20:58:45 UTC Nov 1 2020       Last leave: 20:58:37 UTC Nov 1 2020</pre>

## 시나리오 1

양방향으로 ~4초 이상 CCL 통신 손실이 발생했습니다.

### 실패 전

FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

복구 후(유닛 역할이 변경되지 않음)

FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

### 분석

실패(CCL 통신이 끊김).

The image shows three terminal windows from a Firepower device:

- unit-1-1 Control Unit:** Shows commands like 'clear cluster info trace' and 'clear cap /'. The bottom line indicates a health check failure: 'firepower# Asking unit unit-3-1 to quit because it failed unit health-check.'
- unit-2-1 Data Unit:** Shows a series of 'firepower#' prompts, indicating it is in a standby or monitoring state.
- unit-3-1 Data Unit:** Shows a warning about dynamic routing on management interfaces and a status change: 'Cluster unit unit-3-1 transitioned from SECONDARY to PRIMARY'.

unit-3-1의 데이터 플레인 콘솔 메시지:

```
<#root>
```

```
firepower#
```

```
WARNING: dynamic routing is not supported on management interface when cluster interface-mode is 'spanned'. If dynamic routing is configured on any management interface, please remove it.
```

```
Cluster unit unit-3-1 transitioned from SECONDARY to PRIMARY
```

Cluster disable is performing cleanup..done.  
 All data interfaces have been shutdown due to clustering being disabled.  
 To recover either enable clustering or remove cluster group configuration.

Unit-1-1 클러스터 추적 로그:

<#root>

firepower#

show cluster info trace | include unit-3-1

Nov 02 09:38:14.239 [INFO]Notify chassis de-bundle port for blade unit-3-1, stack 0x000055a8918307fb 0x  
 Nov 02 09:38:14.239 [INFO]FTD - CD proxy received state notification (DISABLED) from unit unit-3-1  
 Nov 02 09:38:14.239

[DEBUG]Send CCP message to all: CCP\_MSG\_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER\_QUIT\_MEMBER\_DR

Nov 02 09:38:14.239 [INFO]Notify chassis de-bundle port for blade unit-3-1, stack 0x000055a8917eb596 0x  
 Nov 02 09:38:14.239

[DEBUG]Send CCP message to id 1: CCP\_MSG\_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER\_QUIT\_REASON\_UN

Nov 02 09:38:14.239 [CRIT]Received heartbeat event 'SECONDARY heartbeat failure' for member unit-3-1 (I

스플릿 브레인

Unit-1-1	Unit-2-1
<pre> &lt;#root&gt; firepower# show cluster info  Cluster GROUP1: On   Interface mode: spanned  This is "unit-1-1" in state PRIMARY        ID       : 0       Site ID  : 1       Version  : 9.12(2)33       Serial No.: FCH22247LNK       CCL IP   : 10.17.1.1       CCL MAC  : 0015.c500.018f       Last join : 20:25:36 UTC Nov 1 2020           </pre>	<pre> &lt;#root&gt; firepower# show cluster info  Cluster GROUP1: On   Interface mode: spanned   This is "unit-2-1" in state S       ID       : 2       Site ID  : 1       Version  : 9.12(2)33       Serial No.: FCH23157Y9N       CCL IP   : 10.17.2.1       CCL MAC  : 0015.c500.028       Last join : 20:44:46 UTC       Last leave: 20:44:38 UTC Other members in the cluster:           </pre>

<pre> Last leave: 20:25:28 UTC Nov 1 2020 Other members in the cluster: Unit "unit-2-1" in state SECONDARY ID      : 2 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH23157Y9N CCL IP  : 10.17.2.1 CCL MAC : 0015.c500.028f Last join : 20:44:45 UTC Nov 1 2020 Last leave: 20:44:38 UTC Nov 1 2020 </pre>	<pre> Unit "unit-1-1" in state PRIMARY  ID      : 0 Site ID : 1 Version : 9.12(2)33 Serial No.: FCH22247LNK CCL IP  : 10.17.1.1 CCL MAC : 0015.c500.018 Last join : 20:25:36 UTC Last leave: 20:25:28 UTC </pre>
--	--

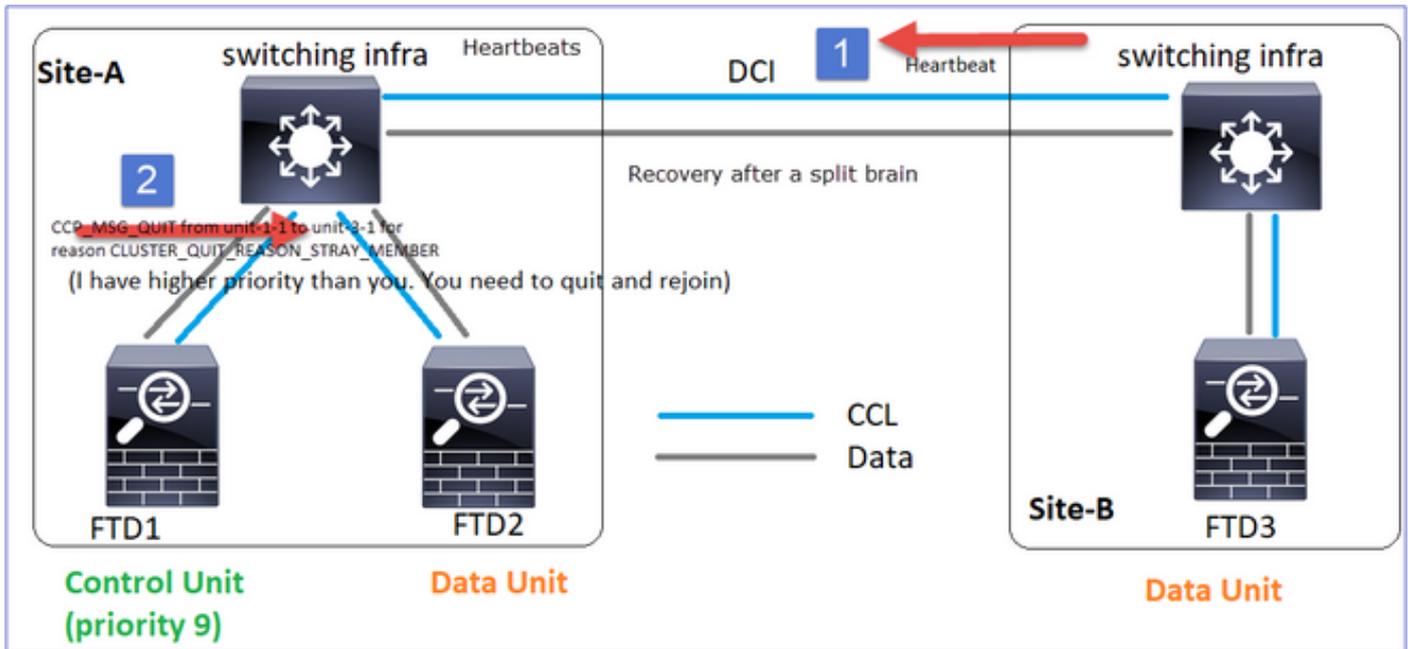
### 클러스터 기록

Unit-1-1	Unit-2-1	Unit-3-1
이벤트 없음	이벤트 없음	<pre> &lt;#root&gt; 09:38:16 UTC Nov 2 2020 SECONDARY PRIMARY_POST_CONFIG Primary relinquished 09:38:17 UTC Nov 2 2020 PRIMARY_POST_CONFIG Primary Primary post config done </pre>

### CCL 통신 복원

Unit-1-1은 현재 제어 노드를 탐지하며, unit-1-1의 우선순위가 더 높기 때문에 유닛-3-1에 CLUSTER\_QUIT\_REASON\_STRAY\_MEMBER 메시지를 보내 새 선택 프로세스를 트리거합니다. 결국 unit-3-1은 데이터 노드로 다시 합류하게 된다.

스플릿 파티션이 피어 파티션과 다시 연결되면, 데이터 노드는 우세 제어 노드에 의해 스트레이 멤버로 취급되며 CLUSTER\_QUIT\_REASON\_STRAY\_MEMBER를 이유로 하는 CCP quit 메시지를 받습니다.



<#root>

Unit-3-1 console logs show:

```
Cluster unit unit-3-1 transitioned from PRIMARY to DISABLED
```

The 3DES/AES algorithms require a Encryption-3DES-AES activation key.

```
Detected Cluster Primart.
```

```
Beginning configuration replication from Primary.
```

```
WARNING: Local user database is empty and there are still 'aaa' commands for 'LOCAL'.
```

```
..
Cryptochecksum (changed): a9ed686f 8e2e689c 2553a104 7a2bd33a
End configuration replication from Primary.
```

```
Cluster unit unit-3-1 transitioned from DISABLED to SECONDARY
```

두 유닛(unit-1-1 및 unit-3-1) 모두 클러스터 로그에 표시됩니다.

<#root>

```
firepower#
```

```
show cluster info trace | include retain
```

```
Nov 03 21:20:23.019 [CRIT]Found a split cluster with both unit-1-1 and unit-3-1 as primary units. Prima
```

```
Nov 03 21:20:23.019 [CRIT]Found a split cluster with both unit-1-1 and unit-3-1 as primary units. Prima
```

스플릿 브레인에 대해 생성되는 syslog 메시지도 있습니다.

```
<#root>
```

```
firepower#
```

```
show log | include 747016
```

```
Nov 03 2020 21:20:23: %FTD-4-747016: Clustering: Found a split cluster with both unit-1-1 and unit-3-1
Nov 03 2020 21:20:23: %FTD-4-747016: Clustering: Found a split cluster with both unit-1-1 and unit-3-1
```

### 클러스터 기록

Unit-1-1	Unit-2-1	Unit-3-1
이벤트 없음	이벤트 없음	<pre>&lt;#root&gt; 09:47:33 UTC Nov 2 2020  Primary DISABLED          Detected a splitted cluster  09:47:38 UTC Nov 2 2020 DISABLED          ELECTION          Enabled from CLI 09:47:38 UTC Nov 2 2020 ELECTION          SECONDARY_COLD          Received cluster control me 09:47:38 UTC Nov 2 2020 SECONDARY_COLD          SECONDARY_APP_SYNC          Client progression done 09:48:18 UTC Nov 2 2020 SECONDARY_APP_SYNC          SECONDARY_CONFIG          SECONDARY application o 09:48:29 UTC Nov 2 2020 SECONDARY_CONFIG          SECONDARY_FILESYS          Configuration replicati 09:48:30 UTC Nov 2 2020 SECONDARY_FILESYS          SECONDARY_BULK_SYNC          Client progression done 09:48:54 UTC Nov 2 2020 SECONDARY_BULK_SYNC  SECONDARY  Client progression done</pre>

### 시나리오 2

양방향으로 ~3-4초 동안 CCL 통신 손실이 발생했습니다.

### 실패 전

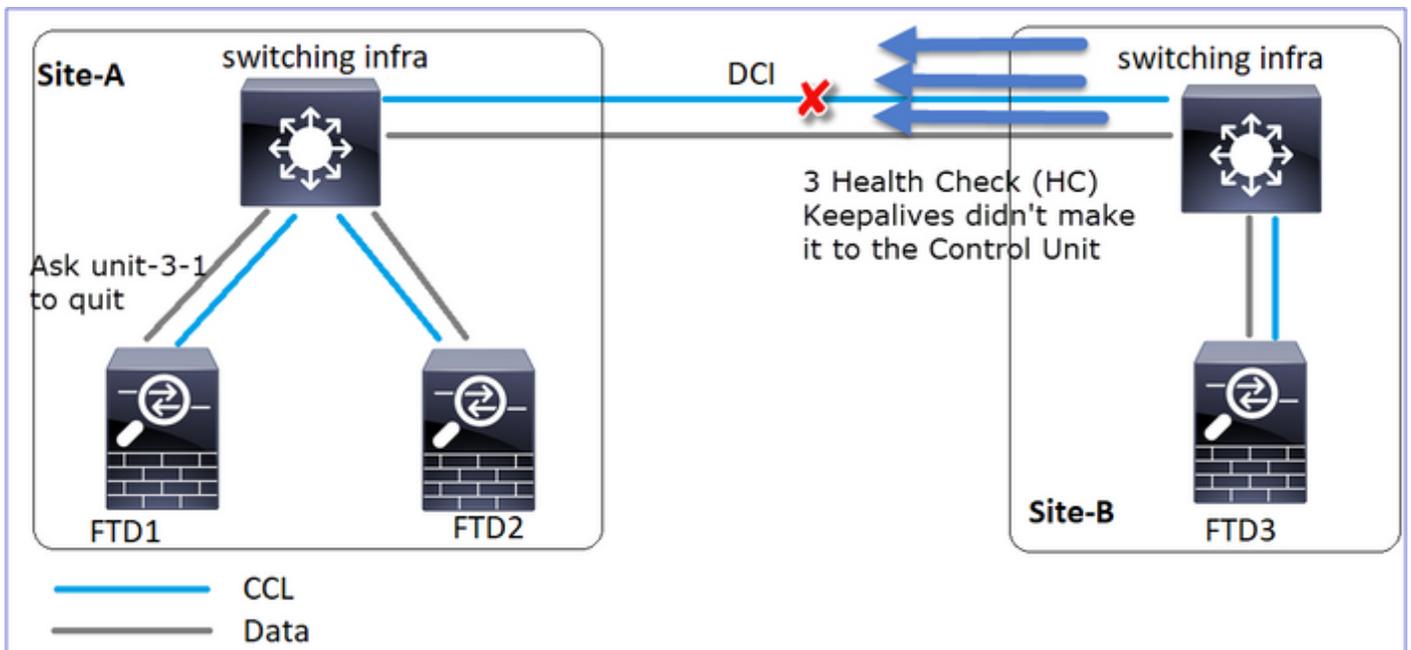
FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

복구 후(유닛 역할이 변경되지 않음)

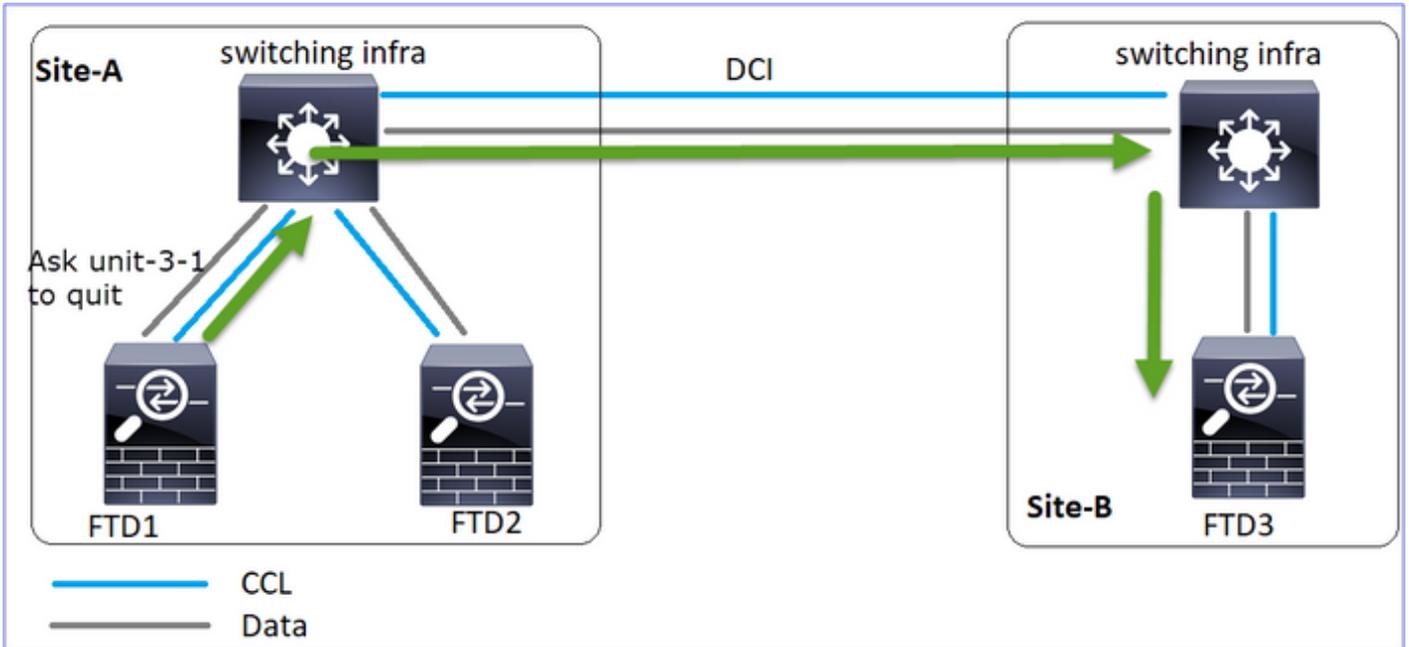
FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

분석

이벤트 1: 제어 노드는 유닛-3-1에서 3 개의 HC를 손실하고 유닛-3-1로 메시지를 보내 클러스터를 떠난다.



이벤트 2: CCL이 매우 빠르게 복구되었고 제어 노드의 CLUSTER\_QUIT\_REASON\_STRAY\_MEMBER 메시지가 원격 측에 전달되었습니다. Unit-3-1은 DISABLED 모드로 직접 전환되며 스플릿 브레인(split-brain)이 없습니다.



unit-1-1(control)에서 다음을 볼 수 있습니다.

```
<#root>
```

```
firepower#
Asking SECONDARY unit unit-3-1 to quit because it failed unit health-check.
```

```
Forcing stray member unit-3-1 to leave the cluster
```

unit-3-1(데이터 노드)에서 다음을 볼 수 있습니다.

```
<#root>
```

```
firepower#
```

```
Cluster disable
```

```
is performing cleanup..done.
All data interfaces have been shutdown due to clustering being disabled. To recover either enable cluster
Cluster unit unit-3-1 transitioned from SECONDARY to DISABLED
```

클러스터 유닛 unit-3-1이 DISABLED 상태로 전환되었으며 CCL 통신이 복원되면 데이터 노드로 다시 연결됩니다.

```
<#root>
```

```
firepower#
```

```
show cluster history
```

20:58:40 UTC Nov 1 2020

SECONDARY                    DISABLED                    Received control message DISABLE (stray member)

20:58:45 UTC Nov 1 2020

DISABLED                    ELECTION                    Enabled from CLI

20:58:45 UTC Nov 1 2020

ELECTION                    SECONDARY\_COLD                    Received cluster control message

20:58:45 UTC Nov 1 2020

SECONDARY\_COLD                    SECONDARY\_APP\_SYNC                    Client progression done

20:59:33 UTC Nov 1 2020

SECONDARY\_APP\_SYNC                    SECONDARY\_CONFIG                    SECONDARY application configuration sync done

20:59:44 UTC Nov 1 2020

SECONDARY\_CONFIG                    SECONDARY\_FILESYS                    Configuration replication finished

20:59:45 UTC Nov 1 2020

SECONDARY\_FILESYS                    SECONDARY\_BULK\_SYNC                    Client progression done

21:00:09 UTC Nov 1 2020

SECONDARY\_BULK\_SYNC                    SECONDARY  
Client progression done

### 시나리오 3

양방향으로 ~3-4초 동안 CCL 통신 손실이 발생했습니다.

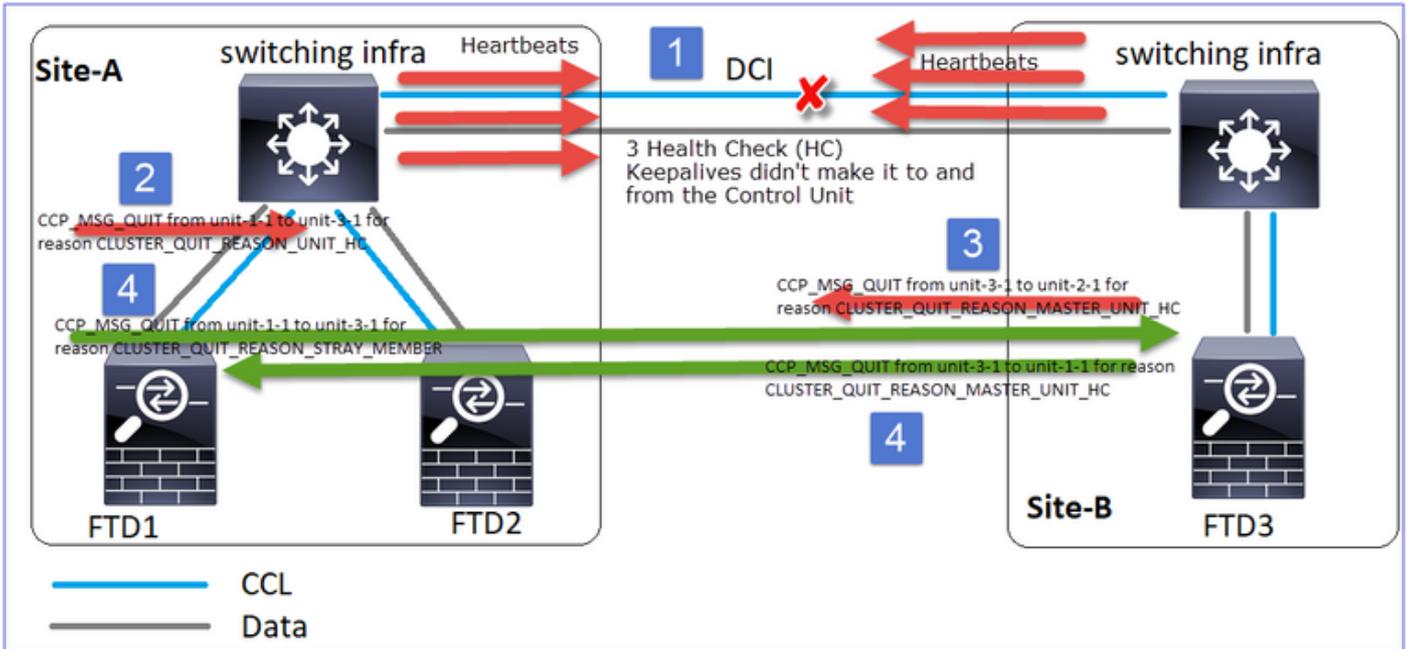
실패하기 전예요

FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

복구 후(제어 노드가 변경됨).

FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
데이터 노드	제어 노드	데이터 노드

## 분석



1. CCL이 다운되었습니다.
2. Unit-1-1은 unit-3-1에서 3개의 HC 메시지를 받지 못하고 unit-3-1로 QUIT 메시지를 보냅니다. 이 메시지는 unit-3-1에 도달하지 않습니다.
3. 장치-3-1은 장치-2-1에 QUIT 메시지를 보냅니다. 이 메시지는 장치-2-1에 도달하지 않습니다.

### CCL 복구.

4. Unit-1-1은 유닛-3-1이 자신을 제어 노드로 광고한 것을 확인하고 QUIT\_REASON\_STRAY\_MEMBER 메시지를 유닛-3-1로 보냅니다. 유닛-3-1이 이 메시지를 DISABLED 상태로 전환하면 됩니다. 그와 동시에 unit-3-1은 QUIT\_REASON\_PRIMARY\_UNIT\_HC 메시지를 unit-1-1로 전송하여 종료를 요청합니다. unit-1-1에서 이 메시지를 가져오면 DISABLED 상태가 됩니다.

### 클러스터 기록

```

Unit-1-1
<#root>
19:53:09 UTC Nov 2 2020
PRIMARY DISABLED
    Received control message DISABLE
                                (primary unit health check failure)
19:53:13 UTC Nov 2 2020
DISABLED          ELECTION          Enabled from CLI
19:53:13 UTC Nov 2 2020
ELECTION         SECONDARY_COLD         Received cluster control message
    
```

```

19:53:13 UTC Nov 2 2020
SECONDARY_COLD          SECONDARY_APP_SYNC      Client progression done
19:54:01 UTC Nov 2 2020
SECONDARY_APP_SYNC      SECONDARY_CONFIG        SECONDARY application configur
19:54:12 UTC Nov 2 2020
SECONDARY_CONFIG        SECONDARY_FILESYS        Configuration replication fini
19:54:13 UTC Nov 2 2020
SECONDARY_FILESYS       SECONDARY_BULK_SYNC      Client progression done
19:54:37 UTC Nov 2 2020
SECONDARY_BULK_SYNC

```

SECONDARY

Client progression done

#### 시나리오 4

~3~4초 동안 CCL 통신 손실

실패 전

FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

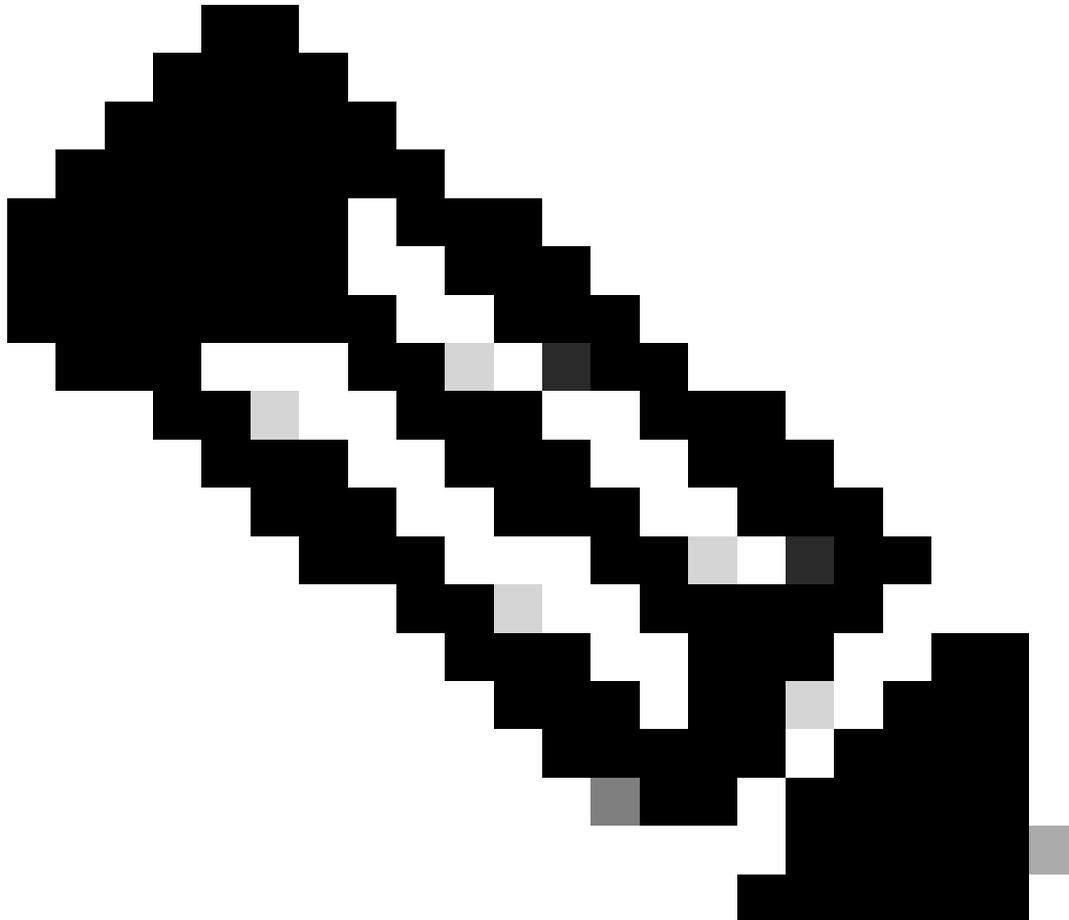
복구 후(제어 노드가 사이트를 변경함)

FTD1	FTD2	FTD3
------	------	------



1은 이를 수신하여 클러스터를 종료합니다.

4. Unit-3-1은 QUIT\_REASON\_PRIMARY\_UNIT\_HC 메시지를 unit-1-1로 전송합니다. Unit-1-1은 이를 수신하여 클러스터를 종료합니다. CCL 복구.
  5. 유닛-1-1 및 유닛-2-1은 데이터 노드로 클러스터에 다시 가입합니다.
- 



참고: 5단계에서 CCL이 복구되지 않으면 site-A에서 FTD1이 새 제어 노드가 되고, CCL 복구 후 새 선택에서 승리합니다.

---

유닛 1-1의 Syslog 메시지:

```
<#root>
```

```
firepower#
```

```
show log | include 747
```

```
Nov 03 2020 23:13:08: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE  
Nov 03 2020 23:13:09: %FTD-4-747015: Clustering: Forcing stray member unit-3-1 to leave the cluster
```

```
Nov 03 2020 23:13:09: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:10: %FTD-4-747015: Clustering: Forcing stray member unit-3-1 to leave the cluster
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering:
```

```
State machine changed from state PRIMARY to DISABLED
```

```
Nov 03 2020 23:13:12: %FTD-7-747006: Clustering: State machine is at state DISABLED
Nov 03 2020 23:13:12: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MY_STATE (sta
Nov 03 2020 23:13:18: %FTD-6-747004: Clustering: State machine changed from state ELECTION to ONCALL
```

### 유닛 1-1의 클러스터 추적 로그:

```
<#root>
```

```
firepower#
```

```
show cluster info trace | include QUIT
```

```
Nov 03 23:13:10.789 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 for reason CLUSTER_QUIT_R
Nov 03 23:13:10.769 [DEBUG]
```

```
Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-1-1 for reason CLUSTER_QUIT_REASON_PRIMARY_UNIT
```

```
Nov 03 23:13:10.769 [DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
Nov 03 23:13:09.789 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_REASON
Nov 03 23:13:09.769 [DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
Nov 03 23:13:08.559 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CL
Nov 03 23:13:08.559 [DEBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
```

### 유닛 3-1의 Syslog 메시지:

```
<#root>
```

```
firepower#
```

```
show log | include 747
```

```
Nov 03 2020 23:13:09: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:10: %FTD-7-747005: Clustering: State machine notify event CLUSTER_EVENT_MEMBER_STATE
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering:
```

```
State machine changed from state SECONDARY to PRIMARY
```

```
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_FAST to PRIMA
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_DRAIN to PRIM
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_CONFIG to PRI
Nov 03 2020 23:13:10: %FTD-7-747006: Clustering: State machine is at state PRIMARY_POST_CONFIG
Nov 03 2020 23:13:10: %FTD-6-747004: Clustering: State machine changed from state PRIMARY_POST_CONFIG t
Nov 03 2020 23:13:10: %FTD-7-747006: Clustering:
```

```
State machine is at state PRIMARY
```

클러스터 기록

Unit-1-1		
<pre> &lt;#root&gt; 23:13:13 UTC Nov 3 2020  PRIMARY DISABLED      Received control message DISABLE (primary unit health check failure)  23:13:18 UTC Nov 3 2020 DISABLED      ELECTION      Enabled from CLI 23:13:18 UTC Nov 3 2020 ELECTION      ONCALL      Received cluster control message 23:13:23 UTC Nov 3 2020 ONCALL      ELECTION      Received cluster control message ... 23:14:48 UTC Nov 3 2020 ONCALL      ELECTION      Received cluster control message 23:14:48 UTC Nov 3 2020 ELECTION      SECONDARY_COLD      Received cluster control message 23:14:48 UTC Nov 3 2020 SECONDARY_COLD      SECONDARY_APP_SYNC      Client progression done 23:15:36 UTC Nov 3 2020 SECONDARY_APP_SYNC      SECONDARY_CONFIG      SECONDARY application configuration sync done 23:15:48 UTC Nov 3 2020 SECONDARY_CONFIG      SECONDARY_FILESYS      Configuration replication finished 23:15:49 UTC Nov 3 2020 SECONDARY_FILESYS      SECONDARY_BULK_SYNC      Client progression done 23:16:13 UTC Nov 3 2020 SECONDARY_BULK_SYNC  SECONDARY  Client progression done                 </pre>		

시나리오 5

실패 전

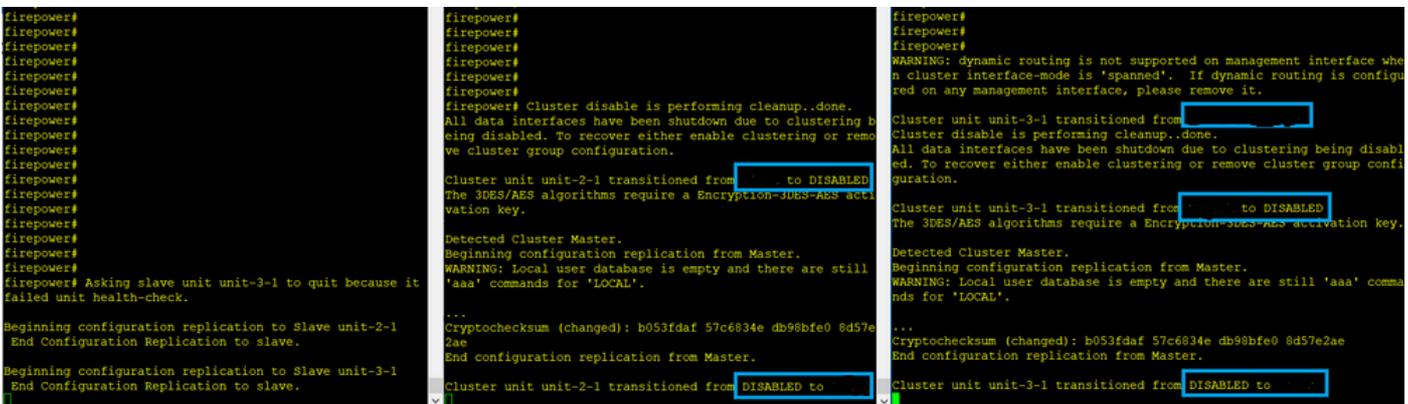
FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B

제어 노드	데이터 노드	데이터 노드
-------	--------	--------

복구 후(변경 사항 없음)

FTD1	FTD2	FTD3
사이트 A	사이트 A	사이트 B
제어 노드	데이터 노드	데이터 노드

### 실패



Unit-3-1은 unit-1-1과 unit-2-1 모두에 QUIT 메시지를 보냈지만 연결 문제로 인해 unit-2-1만 QUIT 메시지를 받았습니다.

Unit-1-1 클러스터 추적 로그:

```
<#root>
```

```
firepower#
```

```
show cluster info trace | include QUIT
```

```
Nov 04 00:52:10.429 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 for reason CLUSTER_QUIT_REASON
Nov 04 00:51:47.059 [DEBUG]Receive CCP message: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_REASON
Nov 04 00:51:45.429 [DEBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER_QUIT_REASON
Nov 04 00:51:45.429 [DEBUG]Send CCP message to unit-3-1(1): CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER_QUIT_REASON
```

Unit-2-1 클러스터 추적 로그:

```
<#root>
```

```
firepower#
```

show cluster info trace | include QUIT

Nov 04 00:52:10.389 [DEBUG]Receive CCP message: CCP\_MSG\_QUIT from unit-3-1 for reason CLUSTER\_QUIT\_REASON...  
Nov 04 00:51:47.019 [DEBUG]Send CCP message to all: CCP\_MSG\_QUIT from unit-2-1 for reason CLUSTER\_QUIT\_REASON...  
Nov 04 00:51:46.999 [DEBUG]

Receive CCP message: CCP\_MSG\_QUIT from unit-3-1 to unit-2-1 for reason CLUSTER\_QUIT\_REASON\_PRIMARY\_UNIT...

Nov 04 00:51:45.389 [DEBUG]Receive CCP message: CCP\_MSG\_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER...

### 클러스터 기록

Unit-1-1	Unit-2-1
이벤트 있음	<pre> &lt;#root&gt; 00:51:50 UTC Nov 4 2020  SECONDARY          DISABLED          Received control message DISABLE (primary unit health check failure)  00:51:54 UTC Nov 4 2020 DISABLED          ELECTION          Enabled from CLI 00:51:54 UTC Nov 4 2020 ELECTION          SECONDARY_COLD          Received cluster control message 00:51:54 UTC Nov 4 2020 SECONDARY_COLD          SECONDARY_APP_SYNC          Client progression done 00:52:42 UTC Nov 4 2020 SECONDARY_APP_SYNC          SECONDARY_CONFIG          SECONDARY application configura sync done 00:52:54 UTC Nov 4 2020 SECONDARY_CONFIG          SECONDARY_FILESYS          Configuration replication finis 00:52:55 UTC Nov 4 2020 SECONDARY_FILESYS          SECONDARY_BULK_SYNC          Client progression done 00:53:19 UTC Nov 4 2020 SECONDARY_BULK_SYNC  SECONDARY  Client progression done </pre>

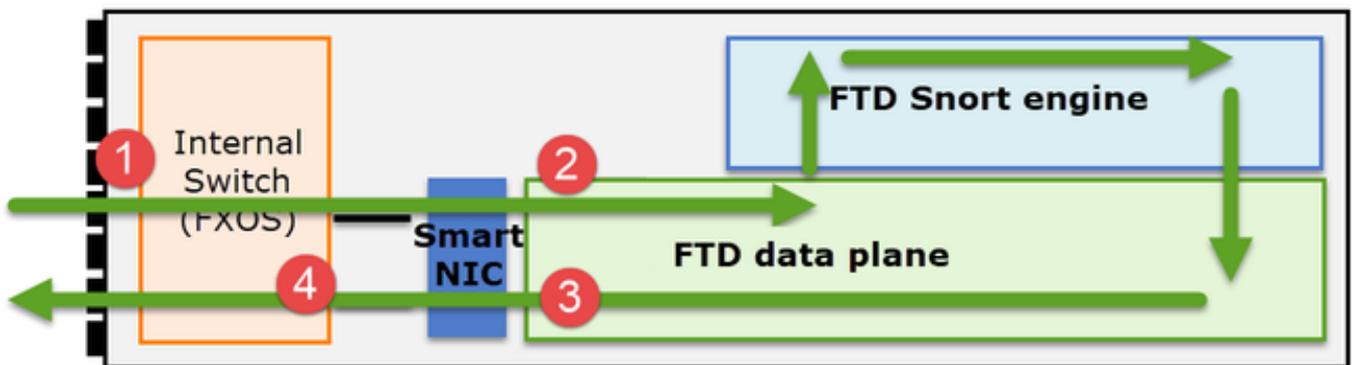
## 클러스터 데이터 플레인 연결 설정

### NGFW 캡처 포인트

NGFW는 다음과 같은 점에서 캡처 기능을 제공합니다.

- 새시 내부 스위치(FXOS)
- FTD 데이터 플레인 엔진
- FTD Snort 엔진

클러스터에서 데이터 경로 문제를 해결할 때 대부분의 경우 사용되는 캡처 지점은 FXOS 및 FTD 데이터 플레인 엔진 캡처입니다.



1. 물리적 인터페이스의 FXOS 인그레스 캡처
2. 데이터 플레인 엔진의 FTD 인그레스 캡처
3. 데이터 플레인 엔진의 FTD 이그레스 캡처
4. 백플레인 인터페이스의 FXOS 인그레스 캡처

NGFW 캡처에 대한 자세한 내용은 다음 문서를 참조하십시오.

### 클러스터 유닛 플로우 역할 기본 사항

클러스터를 통해 다음과 같은 요인에 따라 다양한 방식으로 연결을 설정할 수 있습니다.

- 트래픽 유형(TCP, UDP 등)
- 인접한 스위치에 구성된 로드 밸런싱 알고리즘
- 방화벽에 구성된 기능
- 네트워크 조건(예: IP 프래그먼트화, 네트워크 지연 등)

흐름 역할	설명	플래그
소유자	일반적으로 연결을 처음 수신하는 유닛입니다	UIO

이사	전달자의 소유자 조회 요청을 처리하는 장치입니다.	Y
백업 소유자	이사가 소유자와 동일한 단위가 아닌 한, 이사가 백업 소유자이기도 합니다. 소유자가 자신을 디렉터로 선택하면 별도의 백업 소유자가 선택됩니다.	Y(디렉터도 백업 소유자인 경우) y(디렉터가 백업 소유자가 아닌 경우)
전달자	소유자에게 패킷을 전달하는 유닛	Z
조각 소유자	프래그먼트된 트래픽을 처리하는 유닛	-
새시 백업	새시 간 클러스터에서 디렉터/백업 및 소유자 플로우가 모두 동일한 새시의 유닛에서 소유하는 경우 다른 새시 중 하나의 유닛이 보조 백업/디렉터가 됩니다.  이 역할은 블레이드가 2개 이상인 Firepower 9300 Series의 새시 간 클러스터와 관련이 있습니다.	W

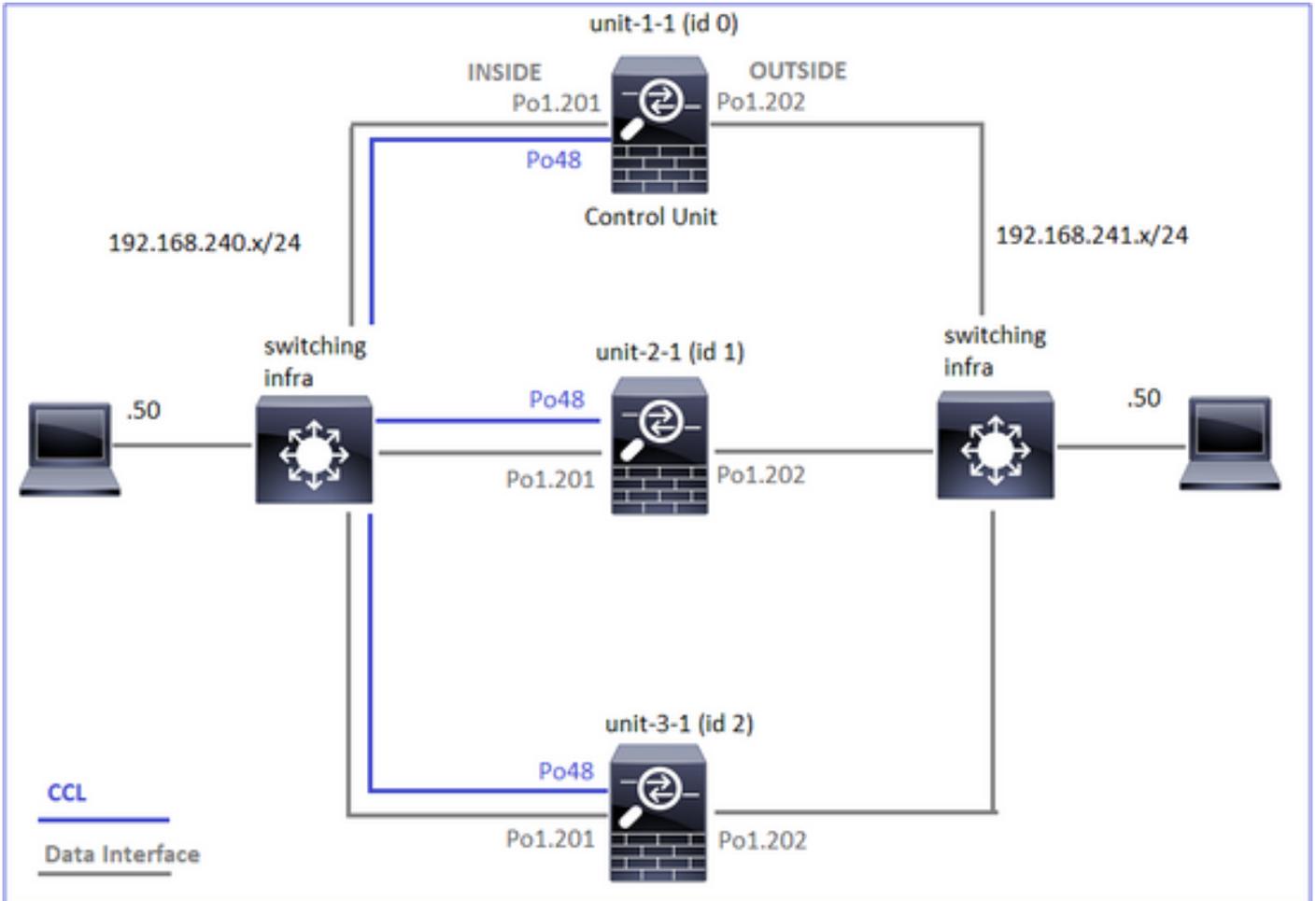
- 자세한 내용은 컨피그레이션 가이드의 관련 섹션을 참조하십시오(관련 정보의 링크 참조).
- 특정 시나리오에서(사례 연구 섹션 참조) 일부 플래그는 항상 표시되지 않습니다.

### 클러스터 연결 설정 사례 연구

다음 섹션에서는 클러스터를 통해 연결을 설정하는 몇 가지 방법을 보여 주는 다양한 사례 연구를 다룹니다. 목표는 다음과 같습니다.

- 다른 유닛 역할에 대해 숙지하십시오.
- 다양한 명령 출력의 상관 관계를 보여 줍니다.

### 토폴로지



클러스터 유닛 및 ID:

Unit-1-1	Unit-2-1
<pre> &lt;#root&gt; Cluster GROUP1: On   Interface mode: spanned  This is "unit-1-1" in state PRIMARY  ID          : 0  Site ID     : 1 Version    : 9.15(1) Serial No.: FCH22247LNK CCL IP     : 10.17.1.1 CCL MAC    : 0015.c500.018f Last join  : 02:24:43 UTC Nov 27 2020 Last leave: N/A           </pre>	<pre> &lt;#root&gt;  Unit "unit-2-1" in state SECO  ID          : 1  Site ID     : 1 Version    : 9.15(1) Serial No.: FCH23157Y9N CCL IP     : 10.17.2.1 CCL MAC    : 0015.c500.02 Last join  : 02:04:19 UTC Last leave: N/A           </pre>

## 클러스터 캡처 사용:

```
cluster exec cap CAPI int INSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CAPO int OUTSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CAPI_RH reinject-hide int INSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CAPO_RH reinject-hide int OUTSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 80
cluster exec cap CCL int cluster buffer 33554432
```

---

 참고: 이러한 테스트는 클러스터를 통과하는 트래픽이 최소화된 랩 환경에서 실행되었습니다. 프로덕션 환경에서는 가능한 한 특정 캡처 필터(예: 대상 포트 및 가능한 경우 소스 포트)로 사용하여 캡처의 '노이즈'를 최소화하려고 합니다.

---

## 사례 연구 1. 대칭 트래픽(소유자도 책임자)

관찰 1. reinject-hide 캡처는 unit-1-1에서만 패킷을 표시합니다. 이는 양방향의 흐름이 unit-1-1을 통과했음을 의미합니다(대칭 트래픽).

<#root>

firepower#

cluster exec show cap

```
unit-1-1(LOCAL):*****
capture CCL type raw-data interface cluster [Capturing - 33513 bytes]
capture CAPI type raw-data buffer 33554432 trace interface INSIDE [Buffer Full - 33553914 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO type raw-data buffer 33554432 trace interface OUTSIDE [Buffer Full - 33553914 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPI_RH type raw-data

reinject-hide

  buffer 33554432 interface INSIDE [Buffer Full -
33553914 bytes

]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO_RH type raw-data

reinject-hide

  buffer 33554432 interface OUTSIDE [Buffer Full -
33553914 bytes

]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
```

```
unit-2-1:*****
capture CCL type raw-data interface cluster [Capturing - 23245 bytes]
capture CAPI type raw-data buffer 33554432 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO type raw-data buffer 33554432 trace interface OUTSIDE [Capturing - 0 bytes]
```

```

match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPI_RH type raw-data

reinject-hide

  buffer 33554432 interface INSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO_RH type raw-data

reinject-hide

  buffer 33554432 interface OUTSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80

unit-3-1:*****
capture CCL type raw-data interface cluster [Capturing - 24815 bytes]
capture CAPI type raw-data buffer 33554432 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO type raw-data buffer 33554432 trace interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPI_RH type raw-data

reinject-hide

  buffer 33554432 interface INSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80
capture CAPO_RH type raw-data

reinject-hide

  buffer 33554432 interface OUTSIDE [Capturing -
0 bytes
]
match tcp host 192.168.240.50 host 192.168.241.50 eq 80

```

## 관찰 2. 소스 포트 45954의 흐름에 대한 연결 플래그 분석

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

```

unit-1-1(LOCAL):*****
22 in use, 25 most used
Cluster:
fwd connections: 0 in use, 1 most used
dir connections: 0 in use, 122 most used

```

centralized connections: 0 in use, 0 most used  
 VPN redirect connections: 0 in use, 0 most used  
 Inspect Snort:  
 preserve-connection: 1 enabled, 0 in effect, 2 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

45954

, idle 0:00:00, bytes 487413076,

flags UIO N1

unit-2-1:\*\*\*\*\*

22 in use, 271 most used

Cluster:

fwd connections: 0 in use, 2 most used

dir connections: 0 in use, 2 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 1 enabled, 0 in effect, 249 most enabled, 0 most in effect

unit-3-1:\*\*\*\*\*

17 in use, 20 most used

Cluster:

fwd connections: 1 in use, 2 most used

dir connections: 1 in use, 127 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:443 NP Identity Ifc 192.168.240.50:39698, idle 0:00:23, bytes 0, flags z

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

45954

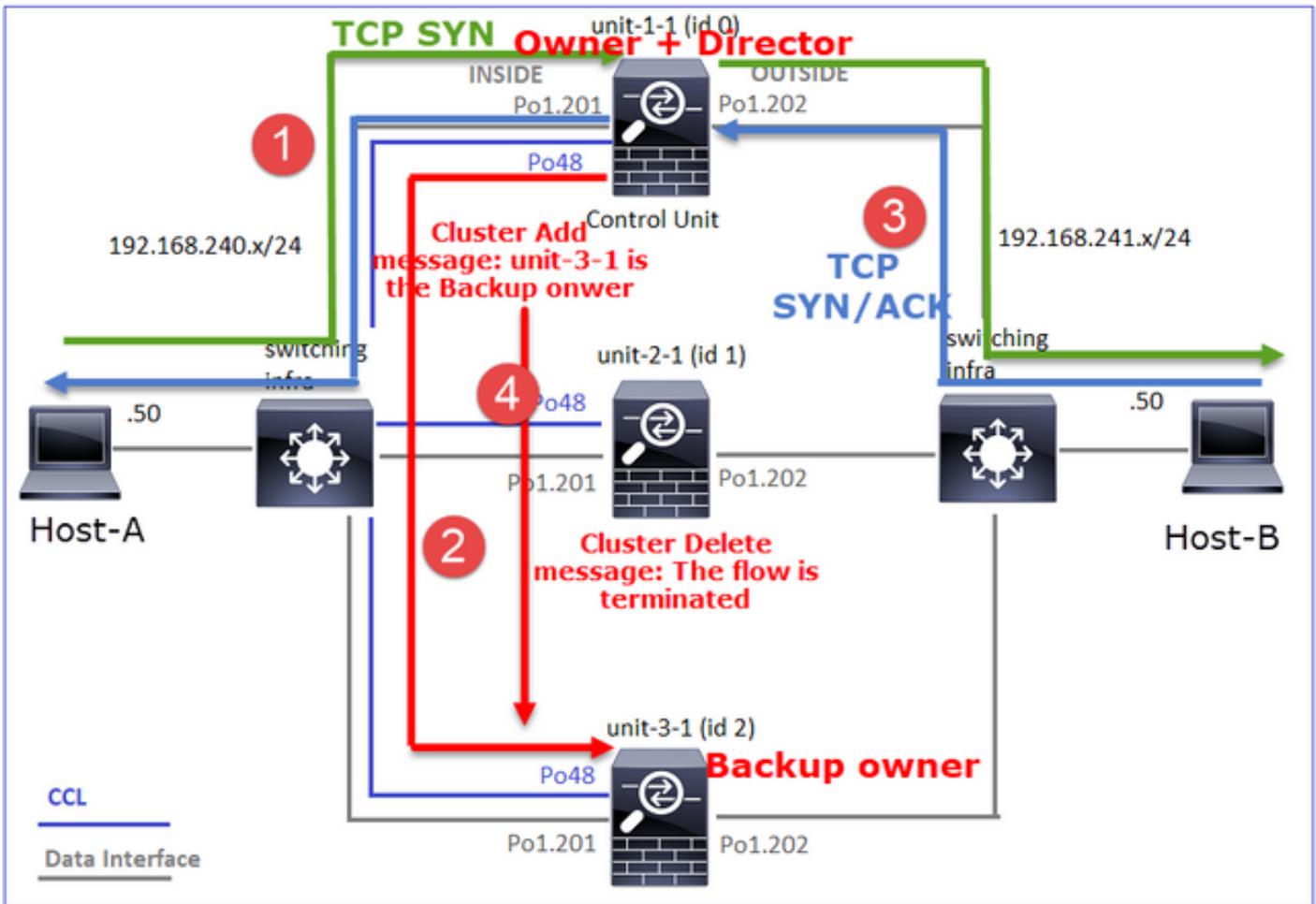
, idle 0:00:06, bytes 0,

flags y

단위	플래그	메모
Unit-1-1	UIO	<ul style="list-style-type: none"> <li>· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.</li> <li>· Director - unit-3-1은 'y'를 갖고 'Y'는 갖고 있지 않으므로 이 플로우의 디렉터로 unit-1-1이 선택되었음을 의미합니다. 그리하여 소유자이기도 하므로 백업 소유자로 다른 유닛(이 사건 unit-3-1)을 선출하였다</li> </ul>
Unit-2-1	-	-

Unit-3-1	y	장치가 백업 소유자입니다.
----------	---	----------------

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-1-1로 도착합니다. Unit-1-1이 흐름 소유자가 됩니다.
2. Unit-1-1도 유동 이사로 선출됩니다. 따라서 유닛 3-1도 백업 소유자(클러스터 추가 메시지)로 선택합니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-3-1로 도착합니다. 흐름은 대칭입니다.
4. 연결이 종료되면 소유자는 클러스터 삭제 메시지를 보내 백업 소유자에서 플로우 정보를 제거합니다.

관찰 3. 흔적으로 캡처하면 양 방향이 유닛-1-1을 통해서만 진행된다는 것을 알 수 있다.

1단계. 소스 포트를 기준으로 모든 클러스터 유닛에서 관심 있는 흐름 및 패킷을 식별합니다.

<#root>

firepower#

```
cluster exec show capture CAPI | i 45954
```

unit-1-1(LOCAL):\*\*\*\*\*

```
1: 08:42:09.362697 802.1Q vlan#201 PO 192.168.240.50.45954 > 192.168.241.50.80: S 992089269:992089269(0
```

```
2: 08:42:09.363521 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.45954: S 4042762409:4042762409
3: 08:42:09.363827 802.1Q vlan#201 PO 192.168.240.50.45954 > 192.168.241.50.80: . ack 4042762410 win 22
...
unit-2-1:*****
unit-3-1:*****
```

<#root>

firepower#

```
cluster exec show capture CAPO | i 45954
```

```
unit-1-1(LOCAL):*****
1: 08:42:09.362987 802.1Q vlan#202 PO 192.168.240.50.45954 > 192.168.241.50.80: S 2732339016:2732339016
2: 08:42:09.363415 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.45954: S 3603655982:3603655982
3: 08:42:09.363903 802.1Q vlan#202 PO 192.168.240.50.45954 > 192.168.241.50.80: . ack 3603655983 win 22
...
unit-2-1:*****
unit-3-1:*****
```

2단계. TCP 흐름 추적이므로 3방향 핸드셰이크 패킷을 추적합니다. 이 출력에서 볼 수 있듯이 unit-1-10이 소유자입니다. 간소화하기 위해 관련되지 않은 추적 단계는 생략됩니다.

<#root>

firepower#

```
show cap CAPI packet-number 1 trace
```

```
25985 packets captured
1: 08:42:09.362697 802.1Q vlan#201 PO 192.168.240.50.
45954
> 192.168.241.50.80:
s
992089269:992089269(0) win 29200 <mss 1460,sackOK,timestamp 495153655 0,nop,wscale 7>
...
Phase: 4
```

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) got initial, attempting ownership.

Phase: 5

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) am becoming owner

...

반환 트래픽(TCP SYN/ACK):

<#root>

firepower#

show capture CAPO packet-number 2 trace

25985 packets captured

2: 08:42:09.363415 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.45954:

S

3603655982:3603655982(0)

ack

2732339017 win 28960 <mss 1460,sackOK,timestamp 505509125 495153655,nop,wscale 7>

...

Phase: 3

Type: FLOW-LOOKUP

Subtype:

Result: ALLOW

Config:

Additional Information:

Found flow with id 9364, using existing flow

관찰 4. FTD 데이터 플레인 syslog는 모든 유닛에서 연결 생성 및 종료를 보여줍니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show log | include 45954
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
Dec 01 2020 08:42:09: %FTD-6-302013:
```

```
Built inbound TCP connection 9364
```

```
for INSIDE:192.168.240.50/45954 (192.168.240.50/45954) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 08:42:18: %FTD-6-302014:
```

```
Teardown TCP connection 9364
```

```
for INSIDE:192.168.240.50/45954 to OUTSIDE:192.168.241.50/80 duration 0:00:08 bytes 1024000440 TCP FIN
```

```
unit-2-1:*****
```

```
unit-3-1
```

```
:*****
```

```
Dec 01 2020 08:42:09: %FTD-6-302022:
```

```
Built backup stub TCP connection
```

```
for INSIDE:192.168.240.50/45954 (192.168.240.50/45954) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 08:42:18: %FTD-6-302023:
```

```
Teardown backup TCP connection
```

```
for INSIDE:192.168.240.50/45954 to OUTSIDE:192.168.241.50/80 duration 0:00:08 forwarded bytes 0 Cluste
```

사례 연구 2. 대칭 트래픽(디렉터와 다른 소유자)

- 사례 연구 #1와 동일하지만, 본 사례 연구에서는 플로우 오너가 디렉터와 다른 단위이다.
- 모든 출력은 사례 연구 #1와 유사합니다. 사례 연구 #1와 비교했을 때 주요 차이점은 시나리오 1의 'y' 플래그를 대체하는 'Y' 플래그입니다.

관찰 1. 소유주가 감독과 다릅니다.

소스 포트 46278의 흐름에 대한 연결 플래그 분석

```
<#root>
```

```
firepower#
```

cluster exec show conn

unit-1-1(LOCAL):\*\*\*\*\*  
23 in use, 25 most used  
Cluster:  
fwd connections: 0 in use, 1 most used  
dir connections: 0 in use, 122 most used  
centralized connections: 0 in use, 0 most used  
VPN redirect connections: 0 in use, 0 most used  
Inspect Snort:  
preserve-connection: 2 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46278

, idle 0:00:00, bytes 508848268, flags

UIO N1

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:46276, idle 0:00:03, bytes 0, flags aA N1

unit-2-1:\*\*\*\*\*  
21 in use, 271 most used  
Cluster:  
fwd connections: 0 in use, 2 most used  
dir connections: 0 in use, 2 most used  
centralized connections: 0 in use, 0 most used  
VPN redirect connections: 0 in use, 0 most used  
Inspect Snort:  
preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

unit-3-1:\*\*\*\*\*  
17 in use, 20 most used  
Cluster:  
fwd connections: 1 in use, 5 most used  
dir connections: 1 in use, 127 most used  
centralized connections: 0 in use, 0 most used  
VPN redirect connections: 0 in use, 0 most used  
Inspect Snort:  
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 NP Identity Ifc 192.168.240.50:46276, idle 0:00:02, bytes 0, flags z

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46278

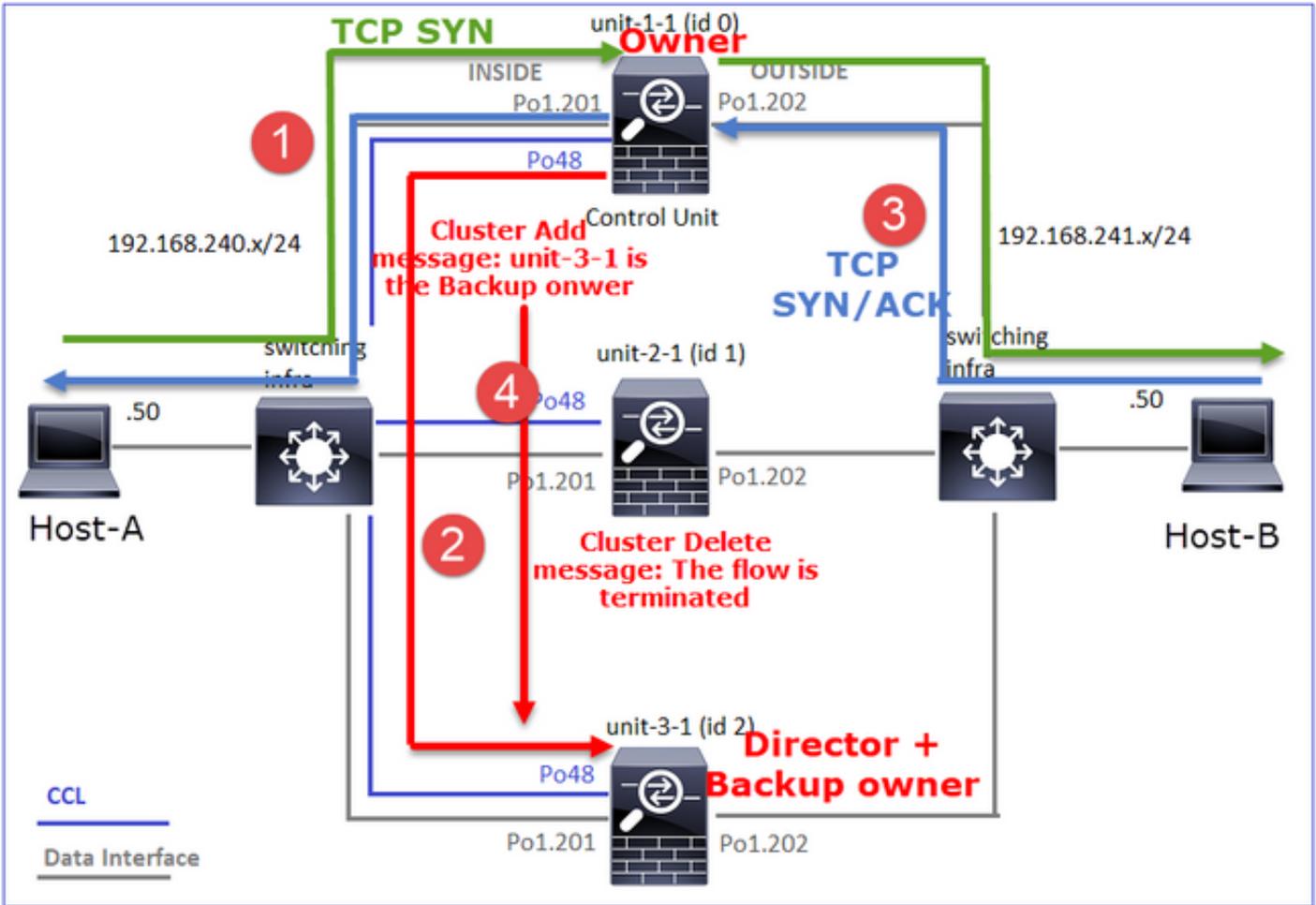
, idle 0:00:06, bytes 0,

flags Y

단위	플래그	메모
Unit-1-1	UIO	· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.

Unit-2-1	-	-
Unit-3-1	Y	· 디렉터 및 백업 소유자 - 유닛 3-1에는 플래그 Y(디렉터)가 있습니다.

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-1-1로 도착합니다. Unit-1-1이 흐름 소유자가 됩니다.
2. Unit-3-1이 플로우 디렉터로 선택됩니다. Unit-3-1도 백업 소유자(CCL을 통한 UDP 4193의 '클러스터 추가' 메시지)입니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-3-1로 도착합니다. 흐름은 대칭입니다.
4. 연결이 종료되면 소유자는 UDP 4193에서 'cluster delete' 메시지를 CCL을 통해 전송하여 백업 소유자의 흐름 정보를 제거합니다.

관찰 2. 흔적으로 캡처하면 양 방향이 유닛-1-1을 통해서만 진행된다는 것을 알 수 있다

1단계. 사례 연구 1과 동일한 접근 방식을 사용하여 소스 포트를 기반으로 모든 클러스터 유닛에서 관심 있는 플로우 및 패킷을 식별합니다.

<#root>

firepower#

```
cluster exec show cap CAPI | include 46278
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
3: 11:01:44.841631 802.1Q vlan#201 PO 192.168.240.50.46278 > 192.168.241.50.80:
```

```
s
```

```
1972783998:1972783998(0) win 29200 <mss 1460,sackOK,timestamp 503529072 0,nop,wscale 7>
```

```
4: 11:01:44.842317 802.1Q vlan#201 PO 192.168.241.50.80 > 192.168.240.50.46278:
```

```
s
```

```
3524167695:3524167695(0)
```

```
ack
```

```
1972783999 win 28960 <mss 1380,sackOK,timestamp 513884542 503529072,nop,wscale 7>
```

```
5: 11:01:44.842592 802.1Q vlan#201 PO 192.168.240.50.46278 > 192.168.241.50.80: . ack 3524167696 win 22
```

```
...
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

```
firepower#
```

외부 인터페이스에서 캡처:

```
<#root>
```

```
firepower#
```

```
cluster exec show cap CAPO | include 46278
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
3: 11:01:44.841921 802.1Q vlan#202 PO 192.168.240.50.46278 > 192.168.241.50.80:
```

```
s
```

```
2153055699:2153055699(0) win 29200 <mss 1380,sackOK,timestamp 503529072 0,nop,wscale 7>
```

```
4: 11:01:44.842226 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46278:
```

```
s
```

```
3382481337:3382481337(0)
```

```
ack
```

```
2153055700 win 28960 <mss 1460,sackOK,timestamp 513884542 503529072,nop,wscale 7>
```

```
5: 11:01:44.842638 802.1Q vlan#202 PO 192.168.240.50.46278 > 192.168.241.50.80: . ack 3382481338 win 22
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

```
firepower#
```

2단계. 인그레스 패킷(TCP SYN 및 TCP SYN/ACK)에 초점을 맞춥니다.

<#root>

firepower#

cluster exec show cap CAPI packet-number 3 trace

unit-1-1(LOCAL):\*\*\*\*\*

824 packets captured

3: 11:01:44.841631 802.1Q vlan#201 P0 192.168.240.50.46278 > 192.168.241.50.80:

S

1972783998:1972783998(0) win 29200 <mss 1460,sackOK,timestamp 503529072 0,nop,wscale 7>

...

Phase: 4

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) got initial, attempting ownership.

Phase: 5

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) am becoming owner

unit-1-1에서 SYN/ACK 추적:

<#root>

firepower#

```
cluster exec show cap CAPO packet-number 4 trace
```

```
unit-1-1(LOCAL):*****
```

```
4: 11:01:44.842226 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.
```

```
46278
```

```
:
```

```
s
```

```
3382481337:3382481337(0)
```

```
ack
```

```
2153055700 win 28960 <mss 1460,sackOK,timestamp 513884542 503529072,nop,wscale 7>
```

```
Phase: 3
```

```
Type: FLOW-LOOKUP
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Found flow with id 9583, using existing flow
```

관찰 3. FTD 데이터 플레인 syslog는 소유자 및 백업 소유자에 대한 연결 생성 및 종료를 보여줍니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show log | include 46278
```

```
unit-1-1(LOCAL):*****
```

```
Dec 01 2020 11:01:44: %FTD-6-302013:
```

```
Built inbound TCP connection
```

```
9583 for INSIDE:192.168.240.50/46278 (192.168.240.50/46278) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 11:01:53: %FTD-6-302014:
```

```
Teardown TCP connection
```

```
9583 for INSIDE:192.168.240.50/46278 to OUTSIDE:192.168.241.50/80 duration 0:00:08 bytes 1024001808 TC
```

```
unit-2-1:*****
```

```
unit-3-1:*****
```

```
Dec 01 2020 11:01:44: %FTD-6-302022:
```

```
Built director stub TCP connection
```

```
for INSIDE:192.168.240.50/46278 (192.168.240.50/46278) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
```

```
Dec 01 2020 11:01:53: %FTD-6-302023:
```

```
Teardown director TCP connection
```

```
for INSIDE:192.168.240.50/46278 to OUTSIDE:192.168.241.50/80 duration 0:00:08 forwarded bytes 0 Cluste
```

### 사례 연구 3. 비대칭 트래픽(디렉터가 트래픽 전달)

관찰 1. reinject-hide는 unit-1-1 및 unit-2-1(비대칭 흐름)의 show 패킷을 캡처합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show cap
```

```
unit-1-1(LOCAL):*****
```

```
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554320 bytes]
```

```
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Buffer Full - 98552 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 98552 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPI_RH type raw-data
```

```
reinject-hide
```

```
buffer 100000 interface
```

```
INSIDE
```

```
[Buffer Full -
```

```
98552 bytes
```

```
]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO_RH type raw-data
```

```
reinject-hide
```

```
buffer 100000 interface
```

```
OUTSIDE
```

```
[Buffer Full -
```

```
99932 bytes
```

```
]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-2-1:*****
```

```
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553268 bytes]
```

```
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99052 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO_RH type raw-data
```

```
reinject-hide
```

```
buffer 100000 interface
```

```
OUTSIDE
```

```
[Buffer Full -
```

99052 bytes

```
]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-3-1:*****
capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 53815 bytes]
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Capturing - 658 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
capture CAPO_RH type raw-data reinject-hide buffer 100000 interface OUTSIDE [Capturing - 658 bytes]
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

## 관찰 2. 소스 포트 46502의 흐름에 대한 연결 플래그 분석

<#root>

firepower#

cluster exec show conn

unit-1-1

(LOCAL):\*\*\*\*\*

23 in use, 25 most used

Cluster:

fwd connections: 0 in use, 1 most used

dir connections: 0 in use, 122 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 2 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46502

, idle 0:00:00, bytes 448760236,

flags UIO N1

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:46500, idle 0:00:06, bytes 0, flags aA N1

unit-2-1

:\*\*\*\*\*

21 in use, 271 most used

Cluster:

fwd connections: 0 in use, 2 most used

dir connections: 1 in use, 2 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46502

, idle 0:00:00, bytes 0,

flags Y

unit-3-1:\*\*\*\*\*

17 in use, 20 most used

Cluster:

fwd connections: 1 in use, 5 most used

dir connections: 0 in use, 127 most used

centralized connections: 0 in use, 0 most used

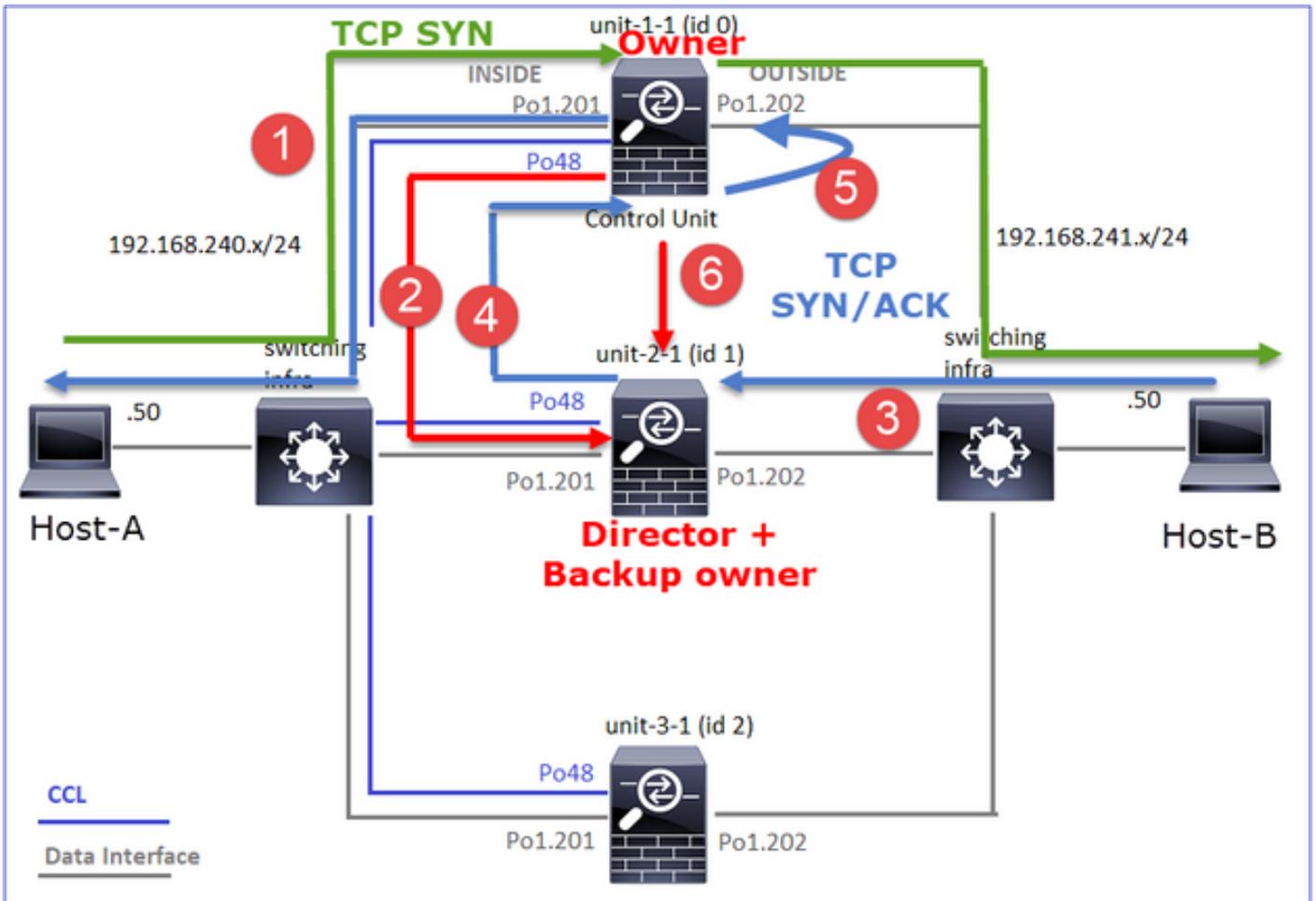
VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

단위	플래그	메모
Unit-1-1	UIO	· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.
Unit-2-1	Y	<ul style="list-style-type: none"> <li>· Director - unit-2-1에는 'Y' 플래그가 있으므로 이 플로우의 Director로 unit-2-1이 선택되었음을 의미합니다.</li> <li>· 백업 소유자</li> <li>· 마지막으로, 이 출력에서 명확하지 않지만 show capture 및 show log 출력에서 유닛-2-1이 소유자에게 이 플로우를 전달하는 것이 분명합니다(이 시나리오에서는 기술적으로 전달자로 간주되지 않지만).</li> </ul> <p>참고: 한 유닛은 디렉터(Y 흐름)와 전달자(z 흐름)가 모두 될 수 없으며, 이 두 역할은 상호 배타적입니다. 디렉터(Y 흐름)는 여전히 트래픽을 전달할 수 있습니다. 이 사례 연구의 뒷부분에 나오는 show log 출력을 참조하십시오.</p>
Unit-3-1	-	-

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-1-1로 도착합니다. Unit-1-1이 흐름 소유자가 됩니다.
2. Unit-2-1은 플로우 디렉터 및 백업 소유자로 선택됩니다. 흐름 소유자가 백업 소유자에게 흐름에 대해 알리기 위해 UDP 4193에서 '클러스터 추가' 유니캐스트 메시지를 보냅니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-2-1로 도착합니다. 흐름이 비대칭적입니다.
4. Unit-2-1은 TCP SYN 쿠키로 인해 패킷을 CCL을 통해 소유자에게 전달합니다.
5. 소유자는 인터페이스 OUTSIDE의 패킷을 다시 거부한 다음 패킷을 Host-A로 전달합니다.
6. 연결이 종료되면 소유자는 클러스터 삭제 메시지를 보내 백업 소유자에서 플로우 정보를 제거합니다.

관찰 3. 추적을 사용하여 캡처하면 비대칭 트래픽과 유닛-2-1에서 유닛-1-1로의 리디렉션이 표시됩니다.

1단계. 관심 흐름에 속하는 패킷을 식별합니다(포트 46502).

```
<#root>
```

```
firepower#
```

```
cluster exec show capture CAPI | include 46502
```

```
unit-1-1(LOCAL):*****
```

```
3: 12:58:33.356121 802.1Q vlan#201 P0 192.168.240.50.46502 > 192.168.241.50.80: S 4124514680:4124514680
```

```
4: 12:58:33.357037 802.1Q vlan#201 P0 192.168.241.50.80 > 192.168.240.50.46502: S 883000451:883000451(0
```

```
5: 12:58:33.357357 802.1Q vlan#201 P0 192.168.240.50.46502 > 192.168.241.50.80: . ack 883000452 win 229
```

```
unit-2-1:*****
```

unit-3-1:\*\*\*\*\*

반환 방향:

<#root>

firepower#

cluster exec show capture CAPO | include 46502

unit-1-1(LOCAL):\*\*\*\*\*

3: 12:58:33.356426 802.1Q vlan#202 PO 192.168.240.50.46502 > 192.168.241.50.80: S 1434968587:1434968587
4: 12:58:33.356915 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: S 4257314722:4257314722
5: 12:58:33.357403 802.1Q vlan#202 PO 192.168.240.50.46502 > 192.168.241.50.80: . ack 4257314723 win 22

unit-2-1:\*\*\*\*\*

1: 12:58:33.359249 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: S 4257314722:4257314722
2: 12:58:33.360302 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: . ack 1434968736 win 23
3: 12:58:33.361004 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.46502: . 4257314723:4257316091
...

unit-3-1:\*\*\*\*\*

2단계. 패킷 추적 기본적으로 처음 50개의 인그레스 패킷만 추적됩니다. 간소화하기 위해 관련되지 않은 추적 단계는 생략됩니다.

Unit-1-1(소유자):

<#root>

firepower#

cluster exec show capture CAPI packet-number 3 trace

unit-1-1(LOCAL):\*\*\*\*\*

3: 12:58:33.356121 802.1Q vlan#201 PO 192.168.240.50.

46502

> 192.168.241.50.80:

s

4124514680:4124514680(0) win 29200 <mss 1460,sackOK,timestamp 510537534 0,nop,wscale 7>

Phase: 4

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'INSIDE'

Flow type: NO FLOW

I (0) got initial, attempting ownership.

Phase: 5  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'INSIDE'  
Flow type: NO FLOW

I (0) am becoming owner

### Unit-2-1(전달자)

반환 트래픽(TCP SYN/ACK) 관심 유닛은 유닛-2-1입니다. 유닛-2-1은 디렉터/백업 소유자이며 트래픽을 소유자에게 전달합니다.

<#root>

firepower#

cluster exec unit unit-2-1 show capture CAPO packet-number 1 trace

1: 12:58:33.359249 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.

46502

: S 4257314722:4257314722(0) ack 1434968588 win 28960 <mss 1460,sackOK,timestamp 520893004 510537534,no

...

Phase: 4  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'OUTSIDE'  
Flow type: NO FLOW

I (1) got initial, attempting ownership.

Phase: 5  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'OUTSIDE'  
Flow type: NO FLOW

I (1) am early redirecting to (0) due to matching action (-1).

관찰 4. FTD 데이터 플레인 syslog는 모든 유닛에서 연결 생성 및 종료를 보여줍니다.

<#root>

firepower#

cluster exec show log | i 46502

unit-1-1(LOCAL):\*\*\*\*\*  
Dec 01 2020 12:58:33: %FTD-6-302013:

B

uilt inbound TCP connection

9742 for INSIDE:192.168.240.50/46502 (192.168.240.50/46502) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)  
Dec 01 2020 12:59:02: %FTD-6-302014:

Teardown TCP connection

9742 for INSIDE:192.168.240.50/46502 to OUTSIDE:192.168.241.50/80 duration 0:00:28 bytes 2048000440 TC

unit-2-1:\*\*\*\*\*  
Dec 01 2020 12:58:33: %FTD-6-302022:

Built forwarder stub TCP connection

for OUTSIDE:192.168.241.50/80 (192.168.241.50/80) to unknown:192.168.240.50/46502 (192.168.240.50/46502)  
Dec 01 2020 12:58:33: %FTD-6-302023:

Teardown forwarder TCP connection

for OUTSIDE:192.168.241.50/80 to unknown:192.168.240.50/46502 duration 0:00:00 forwarded bytes 0 Forwa  
Dec 01 2020 12:58:33: %FTD-6-302022:

Built director stub TCP connection

for INSIDE:192.168.240.50/46502 (192.168.240.50/46502) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)  
Dec 01 2020 12:59:02: %FTD-6-302023:

Teardown director TCP connection

for INSIDE:192.168.240.50/46502 to OUTSIDE:192.168.241.50/80 duration 0:00:28 forwarded bytes 20483163

unit-3-1:\*\*\*\*\*  
firepower#

사례 연구 4. 비대칭 트래픽(소유자가 책임자)

관찰 1. reinject-hide는 unit-1-1 및 unit-2-1(비대칭 흐름)의 show 패킷을 캡처합니다.

<#root>

firepower#

cluster exec show cap

```
unit-1-1(LOCAL):*****  
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554229 bytes]  
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Buffer Full - 98974 bytes]  
match tcp host 192.168.240.50 host 192.168.241.50 eq www  
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 98974 bytes]  
match tcp host 192.168.240.50 host 192.168.241.50 eq www  
capture CAPI_RH type raw-data
```

reinject-hide

buffer 100000 interface

INSIDE

[Buffer Full -

98974 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI\_RH type raw-data

reinject-hide

buffer 100000 interface

OUTSIDE

[Buffer Full -

99924 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

```
unit-2-1:*****
```

capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33552925 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99052 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI\_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO\_RH type raw-data

reinject-hide

buffer 100000 interface OUTSIDE [Buffer Full -

99052 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

```
unit-3-1:*****
```

capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 227690 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Capturing - 4754 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI\_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO\_RH type raw-data reinject-hide buffer 100000 interface OUTSIDE [Capturing - 0 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

## 관찰 2. 소스 포트 46916의 흐름에 대한 연결 플래그 분석

<#root>

firepower#

```
cluster exec show conn
```

unit-1-1

(LOCAL):\*\*\*\*\*

23 in use, 25 most used

Cluster:

fwd connections: 0 in use, 1 most used

dir connections: 0 in use, 122 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 1 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46916

, idle 0:00:00, bytes 414682616,

flags UIO N1

unit-2-1

:\*\*\*\*\*

21 in use, 271 most used

Cluster:

fwd connections: 1 in use, 2 most used

dir connections: 0 in use, 2 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 NP Identity Ifc 192.168.240.50:

46916

, idle 0:00:00, bytes 0,

flags z

unit-3-1

:\*\*\*\*\*

17 in use, 20 most used

Cluster:

fwd connections: 0 in use, 5 most used

dir connections: 1 in use, 127 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

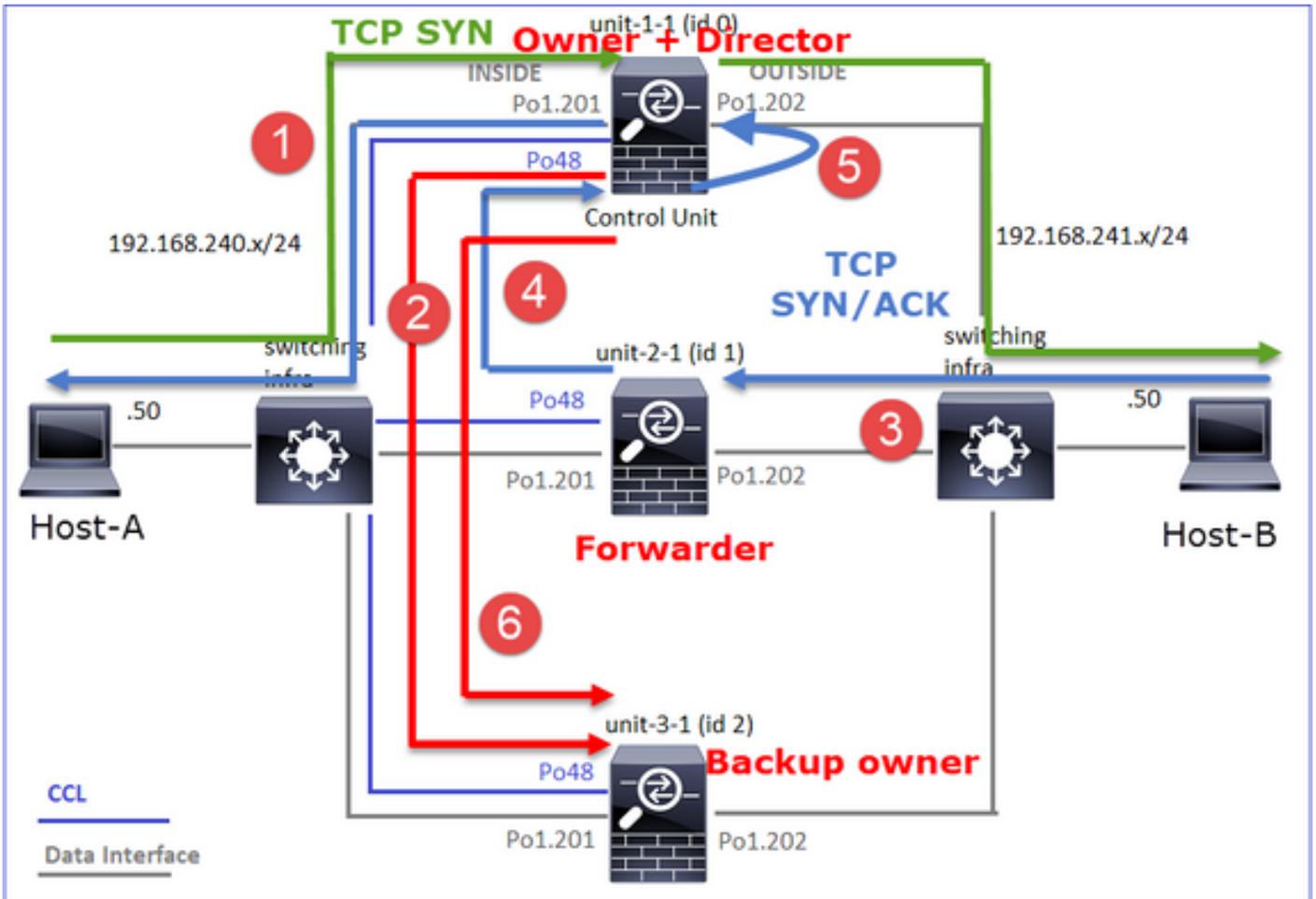
46916

, idle 0:00:04, bytes 0,

flags y

단위	플래그	메모
Unit-1-1	UIO	<ul style="list-style-type: none"><li>· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.</li><li>· Director - unit-3-1은 'y'를 갖고 'Y'는 갖고 있지 않으므로 이 플로우의 디렉터로 unit-1-1이 선택되었음을 의미합니다. 그리하여 소유자이기도 하므로 백업 소유자로 다른 유닛(이 사건 unit-3-1)을 선출하였다</li></ul>
Unit-2-1	z	<ul style="list-style-type: none"><li>· 전달자</li></ul>
Unit-3-1	y	<ul style="list-style-type: none"><li>- 백업 소유자</li></ul>

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-1-1로 전달됩니다. Unit-1-1이 플로우 소유자가 되고 이사가 선택됩니다.
2. Unit-3-1이 백업 소유자로 선택됩니다. 흐름 소유자가 백업 소유자에게 흐름에 대해 알리기 위해 UDP 4193에서 유니캐스트 '클러스터 추가' 메시지를 보냅니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-2-1로 도착합니다. 흐름이 비대칭적입니다.
4. Unit-2-1은 TCP SYN 쿠키로 인해 패킷을 CCL을 통해 소유자에게 전달합니다.
5. 소유자는 인터페이스 OUTSIDE의 패킷을 다시 거부한 다음 패킷을 Host-A로 전달합니다.
6. 연결이 종료되면 소유자는 클러스터 삭제 메시지를 보내 백업 소유자에서 플로우 정보를 제거합니다.

관찰 3. 추적을 사용하여 캡처하면 비대칭 트래픽과 유닛-2-1에서 유닛-1-1로의 리디렉션이 표시됩니다.

Unit-2-1(전달자)

<#root>

firepower#

```
cluster exec unit unit-2-1 show capture CAPO packet-number 1 trace
```

```
1: 16:11:33.653164 802.1Q vlan#202 PO 192.168.241.50.80 > 192.168.240.50.
```

46916

```
:
s
1331019196:1331019196(0)
ack
3089755618 win 28960 <mss 1460,sackOK,timestamp 532473211 522117741,nop,wscale 7>
...
Phase: 4
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW

I (1) got initial, attempting ownership.
```

```
Phase: 5
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW
```

```
I (1) am early redirecting to (0) due to matching action (-1).
```

관찰 4. FTD 데이터 플레인 syslog는 모든 유닛에서 연결 생성 및 종료를 보여줍니다.

- Unit-1-1(소유자)
- Unit-2-1(전달자)
- Unit-3-1(백업 소유자)

```
<#root>
```

```
firepower#
```

```
cluster exec show log | i 46916
```

```
unit-1-1(LOCAL):*****
Dec 01 2020 16:11:33: %FTD-6-302013:
```

```
Built inbound TCP connection
```

```
10023 for INSIDE:192.168.240.50/46916 (192.168.240.50/46916) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)
Dec 01 2020 16:11:42: %FTD-6-302014:
```

```
Teardown TCP connection
```

```
10023 for INSIDE:192.168.240.50/46916 to OUTSIDE:192.168.241.50/80 duration 0:00:09 bytes 1024010016 T
```

unit-2-1:\*\*\*\*\*

Dec 01 2020 16:11:33: %FTD-6-302022:

Built forwarder stub TCP connection

for OUTSIDE:192.168.241.50/80 (192.168.241.50/80) to unknown:192.168.240.50/46916 (192.168.240.50/46916)

Dec 01 2020 16:11:42: %FTD-6-302023:

Teardown forwarder TCP connection

for OUTSIDE:192.168.241.50/80 to unknown:192.168.240.50/46916 duration 0:00:09 forwarded bytes 1024009

unit-3-1:\*\*\*\*\*

Dec 01 2020 16:11:33: %FTD-6-302022:

Built backup stub TCP connection

for INSIDE:192.168.240.50/46916 (192.168.240.50/46916) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)

Dec 01 2020 16:11:42: %FTD-6-302023:

Teardown backup TCP connection

for INSIDE:192.168.240.50/46916 to OUTSIDE:192.168.241.50/80 duration 0:00:09 forwarded bytes 0 Cluste

### 사례 연구 5. 비대칭 트래픽(소유자가 디렉터와 다름)

관찰 1. reinject-hide는 unit-1-1 및 unit-2-1(비대칭 흐름)의 show 패킷을 캡처합니다.

<#root>

firepower#

cluster exec show cap

unit-1-1

(LOCAL):\*\*\*\*\*

capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553207 bytes]

capture CAPI type raw-data buffer 100000 trace interface INSIDE [Buffer Full - 99396 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99224 bytes]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPI\_RH type raw-data

reinject-hide

buffer 100000 interface

INSIDE

[Buffer Full -

99396 bytes

]

match tcp host 192.168.240.50 host 192.168.241.50 eq www

capture CAPO\_RH type raw-data

reinject-hid

```
e buffer 100000 interface
```

```
OUTSIDE
```

```
[Buffer Full -
```

```
99928 bytes
```

```
]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-2-1
```

```
:*****
```

```
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554251 bytes]
```

```
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Buffer Full - 99052 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO_RH type raw-data
```

```
reinject-hide
```

```
buffer 100000 interface
```

```
OUTSIDE
```

```
[Buffer Full -
```

```
99052 bytes
```

```
]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
unit-3-1:*****
```

```
capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 131925 bytes]
```

```
capture CAPI type raw-data buffer 100000 trace interface INSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO type raw-data buffer 100000 trace interface OUTSIDE [Capturing - 2592 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPI_RH type raw-data reinject-hide buffer 100000 interface INSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

```
capture CAPO_RH type raw-data reinject-hide buffer 100000 interface OUTSIDE [Capturing - 0 bytes]
```

```
match tcp host 192.168.240.50 host 192.168.241.50 eq www
```

관찰 2. 소스 포트 46994의 흐름에 대한 연결 플래그 분석:

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

```
unit-1-1
```

(LOCAL):\*\*\*\*\*

23 in use, 25 most used

Cluster:

fwd connections: 0 in use, 1 most used

dir connections: 0 in use, 122 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 1 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

46994

, idle 0:00:00, bytes 406028640,

flags UIO N1

unit-2-1

:\*\*\*\*\*

22 in use, 271 most used

Cluster:

fwd connections: 1 in use, 2 most used

dir connections: 0 in use, 2 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 NP Identity Ifc 192.168.240.50:

46994

, idle 0:00:00, bytes 0,

flags z

unit-3-1

:\*\*\*\*\*

17 in use, 20 most used

Cluster:

fwd connections: 2 in use, 5 most used

dir connections: 1 in use, 127 most used

centralized connections: 0 in use, 0 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.241.50:80 INSIDE 192.168.240.50:

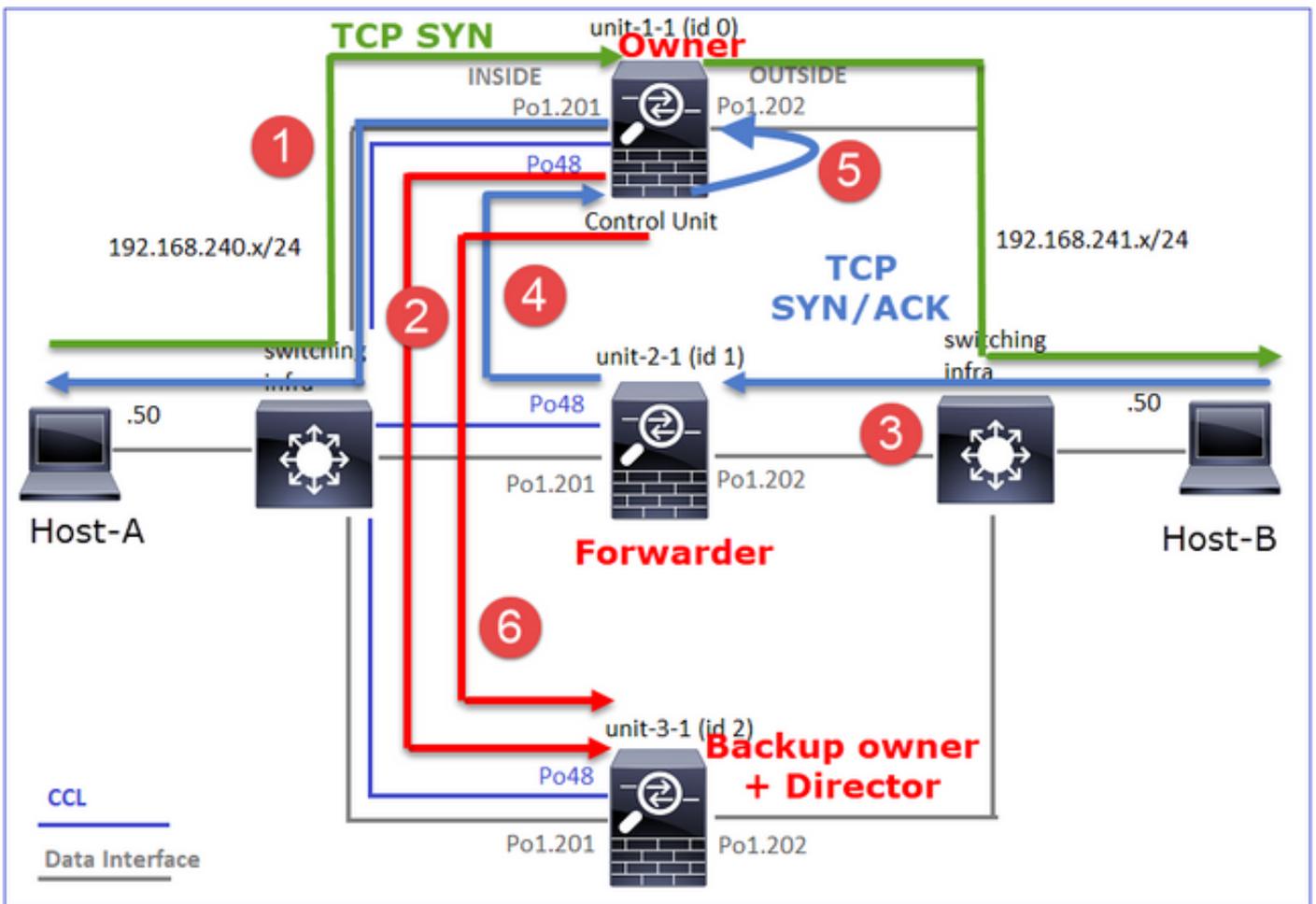
46994

, idle 0:00:05, bytes 0,

flags Y

단위	플래그	메모
Unit-1-1	UIO	· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.
Unit-2-1	z	· 전달자
Unit-3-1	Y	· 백업 소유자 · 디렉터

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-1-1로 도착합니다. Unit-1-1이 흐름 소유자가 됩니다.
2. Unit-3-1은 이사 및 백업 소유자로 선출됩니다. 흐름 소유자가 백업 소유자에게 흐름에 대해 알리기 위해 UDP 4193에서 '클러스터 추가' 유니캐스트 메시지를 보냅니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-2-1로 도착합니다. 흐름은 비대칭입니다
4. Unit-2-1은 TCP SYN 쿠키로 인해 패킷을 CCL을 통해 소유자에게 전달합니다.
5. 소유자는 인터페이스 OUTSIDE의 패킷을 다시 거부한 다음 패킷을 Host-A로 전달합니다.
6. 연결이 종료되면 소유자는 클러스터 삭제 메시지를 보내 백업 소유자에서 플로우 정보를 제거합니다.

관찰 3. 추적을 사용하여 캡처하면 비대칭 트래픽과 유닛-2-1에서 유닛-1-1로의 리디렉션이 표시됩니다.

Unit-1-1(소유자)

<#root>

firepower#

cluster exec show cap CAPI packet-number 1 trace

unit-1-1(LOCAL):\*\*\*\*\*

...  
Phase: 4  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'INSIDE'  
Flow type: NO FLOW

I (0) got initial, attempting ownership.

Phase: 5  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'INSIDE'  
Flow type: NO FLOW

I (0) am becoming owner

Unit-2-1(전달자)

<#root>

firepower#

cluster exec unit unit-2-1 show cap CAPO packet-number 1 trace

1: 16:46:44.232074 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.

46994

: S 2863659376:2863659376(0) ack 2879616990 win 28960 <mss 1460,sackOK,timestamp 534583774 524228304,no

...  
Phase: 4  
Type: CLUSTER-EVENT  
Subtype:

Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'OUTSIDE'  
Flow type: NO FLOW

I (1) got initial, attempting ownership.

Phase: 5  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'OUTSIDE'  
Flow type: NO FLOW

I (1) am early redirecting to (0) due to matching action (-1).

관찰 4. FTD 데이터 플레인 syslog는 모든 유닛에서 연결 생성 및 종료를 보여줍니다.

- Unit-1-1(소유자)
- Unit-2-1(전달자)
- Unit-3-1(백업 소유자/책임자)

<#root>

firepower#

cluster exec show log | i 46994

unit-1-1(LOCAL):\*\*\*\*\*  
Dec 01 2020 16:46:44: %FTD-6-302013:

Built inbound TCP connection

10080 for INSIDE:192.168.240.50/46994 (192.168.240.50/46994) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)  
Dec 01 2020 16:46:53: %FTD-6-302014:

Teardown TCP connection

10080 for INSIDE:192.168.240.50/46994 to OUTSIDE:192.168.241.50/80 duration 0:00:09 bytes 1024000440 T

unit-2-1:\*\*\*\*\*  
Dec 01 2020 16:46:44: %FTD-6-302022:

Built forwarder stub TCP connection

for OUTSIDE:192.168.241.50/80 (192.168.241.50/80) to unknown:192.168.240.50/46994 (192.168.240.50/46994)  
Dec 01 2020 16:46:53: %FTD-6-302023:

Teardown forwarder TCP connection

for OUTSIDE:192.168.241.50/80 to unknown:192.168.240.50/46994 duration 0:00:09 forwarded bytes 1024000

unit-3-1:\*\*\*\*\*

Dec 01 2020 16:46:44: %FTD-6-302022:

Built director stub TCP connection

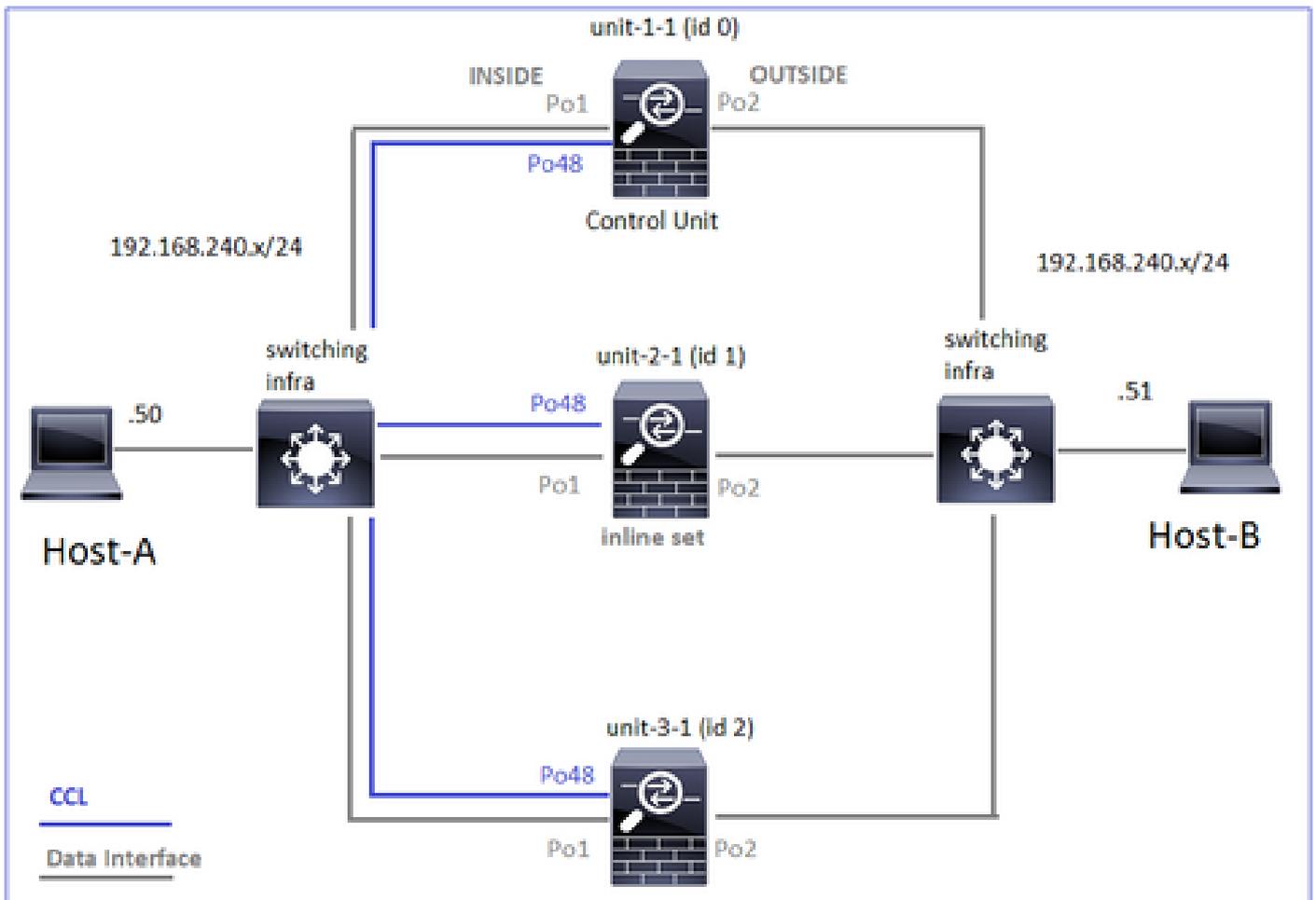
for INSIDE:192.168.240.50/46994 (192.168.240.50/46994) to OUTSIDE:192.168.241.50/80 (192.168.241.50/80)

Dec 01 2020 16:46:53: %FTD-6-302023:

Teardown director TCP connection

for INSIDE:192.168.240.50/46994 to OUTSIDE:192.168.241.50/80 duration 0:00:09 forwarded bytes 0 Cluste

다음 사례 연구에서는 인라인 집합이 있는 클러스터를 기반으로 하는 토폴로지를 사용합니다.



사례 연구 6. 비대칭 트래픽(Inline-set, 소유자가 디렉터)

관찰 1. reinject-hide는 unit-1-1 및 unit-2-1(비대칭 흐름)의 show 패킷을 캡처합니다. 또한 소유자는 unit-2-1입니다(reinject-hide 캡처에 대해 INSIDE 및 OUTSIDE 인터페이스 모두에 패킷이 있는 반면, unit-1-1에는 OUTSIDE에만 패킷이 있습니다).

<#root>

firepower#

cluster exec show cap

unit-1-1

```
(LOCAL):*****  
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553253 bytes]  
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523432 bytes]  
match tcp host 192.168.240.50 host 192.168.240.51 eq www  
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]  
match tcp host 192.168.240.50 host 192.168.240.51 eq www  
capture CAPO_RH type raw-data
```

reinject-hide

interface

OUTSIDE

[Buffer Full -

523432 bytes

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

unit-2-1

```
:*****  
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33554312 bytes]  
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523782 bytes]  
match tcp host 192.168.240.50 host 192.168.240.51 eq www  
capture CAPI type raw-data trace interface INSIDE [Buffer Full - 523782 bytes]  
match tcp host 192.168.240.50 host 192.168.240.51 eq www  
capture CAPO_RH type raw-data
```

reinject-hide

interface

OUTSIDE

[Buffer Full -

524218 bytes

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data
```

reinject-hide

interface

INSIDE

[Buffer Full -

523782 bytes]

```
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

```
unit-3-1:*****  
capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 53118 bytes]  
capture CAPO type raw-data trace interface OUTSIDE [Capturing - 0 bytes]  
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

```
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data reinject-hide interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

## 관찰 2. 소스 포트 51844의 흐름에 대한 연결 플래그 분석

```
<#root>
```

```
firepower#
```

```
cluster exec show conn addr 192.168.240.51
```

```
unit-1-1
```

```
(LOCAL):*****
```

```
30 in use, 102 most used
```

```
Cluster:
```

```
fwd connections: 1 in use, 1 most used
```

```
dir connections: 2 in use, 122 most used
```

```
centralized connections: 3 in use, 39 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

```
preserve-connection: 0 enabled, 0 in effect, 4 most enabled, 1 most in effect
```

```
TCP OUTSIDE 192.168.240.51:80 NP Identity Ifc 192.168.240.50:
```

```
51844
```

```
, idle 0:00:00, bytes 0,
```

```
flags z
```

```
unit-2-1
```

```
:*****
```

```
23 in use, 271 most used
```

```
Cluster:
```

```
fwd connections: 0 in use, 2 most used
```

```
dir connections: 4 in use, 26 most used
```

```
centralized connections: 0 in use, 14 most used
```

```
VPN redirect connections: 0 in use, 0 most used
```

```
Inspect Snort:
```

```
preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect
```

```
TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:
```

```
51844
```

```
, idle 0:00:00, bytes 231214400,
```

```
flags b N
```

unit-3-1

:\*\*\*\*\*

20 in use, 55 most used

Cluster:

fwd connections: 0 in use, 5 most used

dir connections: 1 in use, 127 most used

centralized connections: 0 in use, 24 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

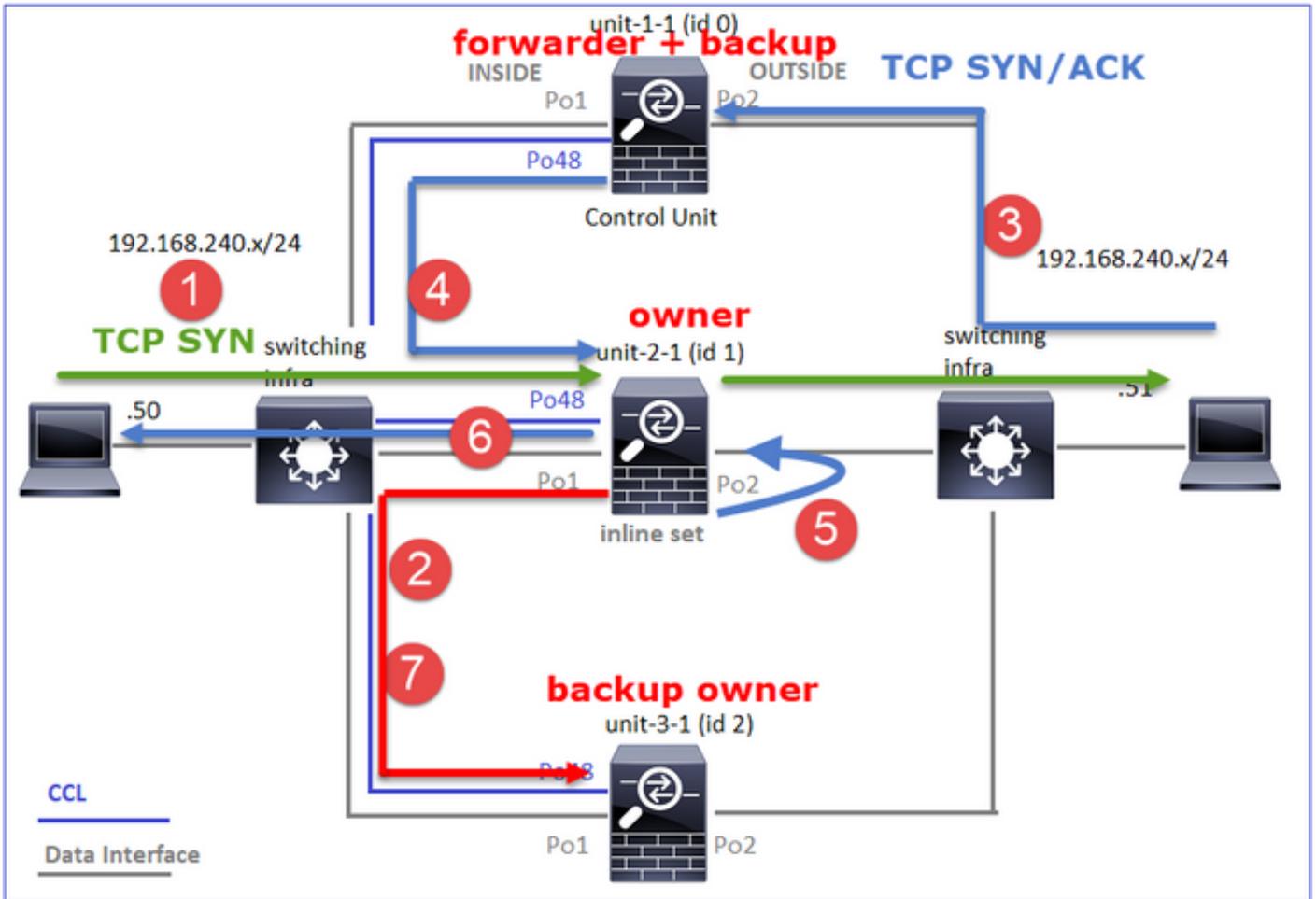
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:51844, idle 0:00:01, bytes 0,

flags y

단위	플래그	메모
Unit-1-1	z	· 전달자
Unit-2-1	b N	· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.
Unit-3-1	y	· 백업 소유자

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-2-1로 전달됩니다. Unit-2-1이 플로우 소유자가 되고 디렉터로 선택됩니다.
2. Unit-3-1이 백업 소유자로 선택됩니다. 흐름 소유자가 백업 소유자에게 흐름에 대해 알리기 위해 UDP 4193에서 '클러스터 추가' 유니캐스트 메시지를 보냅니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-1-1로 도착합니다. 흐름이 비대칭적입니다.
4. Unit-1-1은 CCL을 통해 패킷을 디렉터(unit-2-1)로 전달한다.
5. Unit-2-1도 소유자이며 인터페이스 OUTSIDE의 패킷을 다시 삽입합니다.
6. Unit-2-1은 패킷을 Host-A로 전달합니다.
7. 연결이 종료되면 소유자는 클러스터 삭제 메시지를 보내 백업 소유자에서 플로우 정보를 제거합니다.

관찰 3. 추적을 사용하여 캡처하면 비대칭 트래픽과 유닛-1-1에서 유닛-2-1로의 리디렉션이 표시됩니다.

Unit-2-1(소유자/책임자)

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-2-1 show cap CAPI packet-number 1 trace
```

```
1: 18:10:12.842912 192.168.240.50.51844 > 192.168.240.51.80:
```

```
s
```

```
4082593463:4082593463(0) win 29200 <mss 1460,sackOK,timestamp 76258053 0,nop,wscale 7>
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW
```

I (1) got initial, attempting ownership.

```
Phase: 2
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW
```

I (1) am becoming owner

Unit-1-1(전달자)

<#root>

firepower#

cluster exec show cap CAPO packet-number 1 trace

unit-1-1(LOCAL):\*\*\*\*\*

```
1: 18:10:12.842317 192.168.240.51.80 > 192.168.240.50.51844: S 2339579109:2339579109(0) ack 4082593464
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW
```

I (0) am asking director (1).

반환 트래픽(TCP SYN/ACK)

Unit-2-1(소유자/책임자)

<#root>

firepower#

cluster exec unit unit-2-1 show cap CAPO packet-number 2 trace

2: 18:10:12.843660 192.168.240.51.80 > 192.168.240.50.51844: S 2339579109:2339579109(0) ack 4082593464  
Phase: 1  
Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'OUTSIDE'  
Flow type: FULL

I (1) am owner, update sender (0).

Phase: 2  
Type: FLOW-LOOKUP  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:

Found flow with id 7109, using existing flow

관찰 4. FTD 데이터 플레인 syslog는 모든 유닛에서 연결 생성 및 종료를 보여줍니다.

- Unit-1-1(소유자)
- Unit-2-1(전달자)
- Unit-3-1(백업 소유자/책임자)

<#root>

firepower#

cluster exec show log | include 51844

unit-1-1(LOCAL):\*\*\*\*\*

Dec 02 2020 18:10:12: %FTD-6-302022:

Built forwarder stub TCP connection

for OUTSIDE:192.168.240.51/80 (192.168.240.51/80) to unknown:192.168.240.50/51844 (192.168.240.50/51844)

Dec 02 2020 18:10:22: %FTD-6-302023:

Teardown forwarder TCP connection

for OUTSIDE:192.168.240.51/80 to unknown:192.168.240.50/51844 duration 0:00:09 forwarded bytes 1024001

unit-2-1:\*\*\*\*\*

Dec 02 2020 18:10:12: %FTD-6-302303:

Built TCP state-bypass connection

```
7109 from INSIDE:192.168.240.50/51844 (192.168.240.50/51844) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)
Dec 02 2020 18:10:22: %FTD-6-302304:
```

```
Teardown TCP state-bypass connection
```

```
7109 from INSIDE:192.168.240.50/51844 to OUTSIDE:192.168.240.51/80 duration 0:00:09 bytes 1024001888 T
```

```
unit-3-1:*****
Dec 02 2020 18:10:12: %FTD-6-302022:
```

```
Built backup stub TCP connection
```

```
for INSIDE:192.168.240.50/51844 (192.168.240.50/51844) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)
Dec 02 2020 18:10:22: %FTD-6-302023:
```

```
Teardown backup TCP connection
```

```
for INSIDE:192.168.240.50/51844 to OUTSIDE:192.168.240.51/80 duration 0:00:09 forwarded bytes 0 Cluste
```

### 사례 연구 7. 비대칭 트래픽(Inline-set, 소유자가 디렉터와 다름)

소유자는 unit-2-1입니다(reinject-hide 캡처에 대해 INSIDE 및 OUTSIDE 인터페이스에 패킷이 있는 반면, unit-3-1에는 OUTSIDE에만 패킷이 있습니다).

```
<#root>
```

```
firepower#
```

```
cluster exec show cap
```

```
unit-1-1(LOCAL):*****
capture CCL type raw-data buffer 33554432 interface cluster [Capturing - 13902 bytes]
capture CAPO type raw-data trace interface OUTSIDE [Capturing - 90 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data reinject-hide interface OUTSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

```
unit-2-1
```

```
:*****
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553936 bytes]
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523126 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI type raw-data trace interface INSIDE [Buffer Full - 523126 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data
```

```
reinject-hid
```

```
e
```

```
interface
```

```
OUTSIDE
```

```
[Buffer Full -
524230 bytes
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data
```

```
reinject-hide
```

```
interface
```

```
INSIDE
```

```
[Buffer Full -
```

```
523126 bytes
```

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

```
unit-3-1
```

```
:*****
capture CCL type raw-data buffer 33554432 interface cluster [Buffer Full - 33553566 bytes]
capture CAPO type raw-data trace interface OUTSIDE [Buffer Full - 523522 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPO_RH type raw-data
```

```
reinject-hide
```

```
interface
```

```
OUTSIDE
```

```
[Buffer Full -
```

```
523432 bytes
```

```
]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
capture CAPI_RH type raw-data reinject-hide interface INSIDE [Capturing - 0 bytes]
match tcp host 192.168.240.50 host 192.168.240.51 eq www
```

## 관찰 2. 소스 포트 59210의 흐름에 대한 연결 플래그 분석

```
<#root>
```

```
firepower#
```

```
cluster exec show conn addr 192.168.240.51
```

```
unit-1-1
```

```
(LOCAL):*****
25 in use, 102 most used
```

Cluster:

fwd connections: 0 in use, 1 most used

dir connections: 2 in use, 122 most used

centralized connections: 0 in use, 39 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 4 most enabled, 1 most in effect

TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:

59210

, idle 0:00:03, bytes 0,

flags Y

unit-2-1

:\*\*\*\*\*

21 in use, 271 most used

Cluster:

fwd connections: 0 in use, 2 most used

dir connections: 0 in use, 28 most used

centralized connections: 0 in use, 14 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 249 most enabled, 0 most in effect

TCP OUTSIDE 192.168.240.51:80 INSIDE 192.168.240.50:

59210

, idle 0:00:00, bytes 610132872,

flags b N

unit-3-1

:\*\*\*\*\*

19 in use, 55 most used

Cluster:

fwd connections: 1 in use, 5 most used

dir connections: 0 in use, 127 most used

centralized connections: 0 in use, 24 most used

VPN redirect connections: 0 in use, 0 most used

Inspect Snort:

preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect

TCP OUTSIDE 192.168.240.51:80 NP Identity Ifc 192.168.240.50:

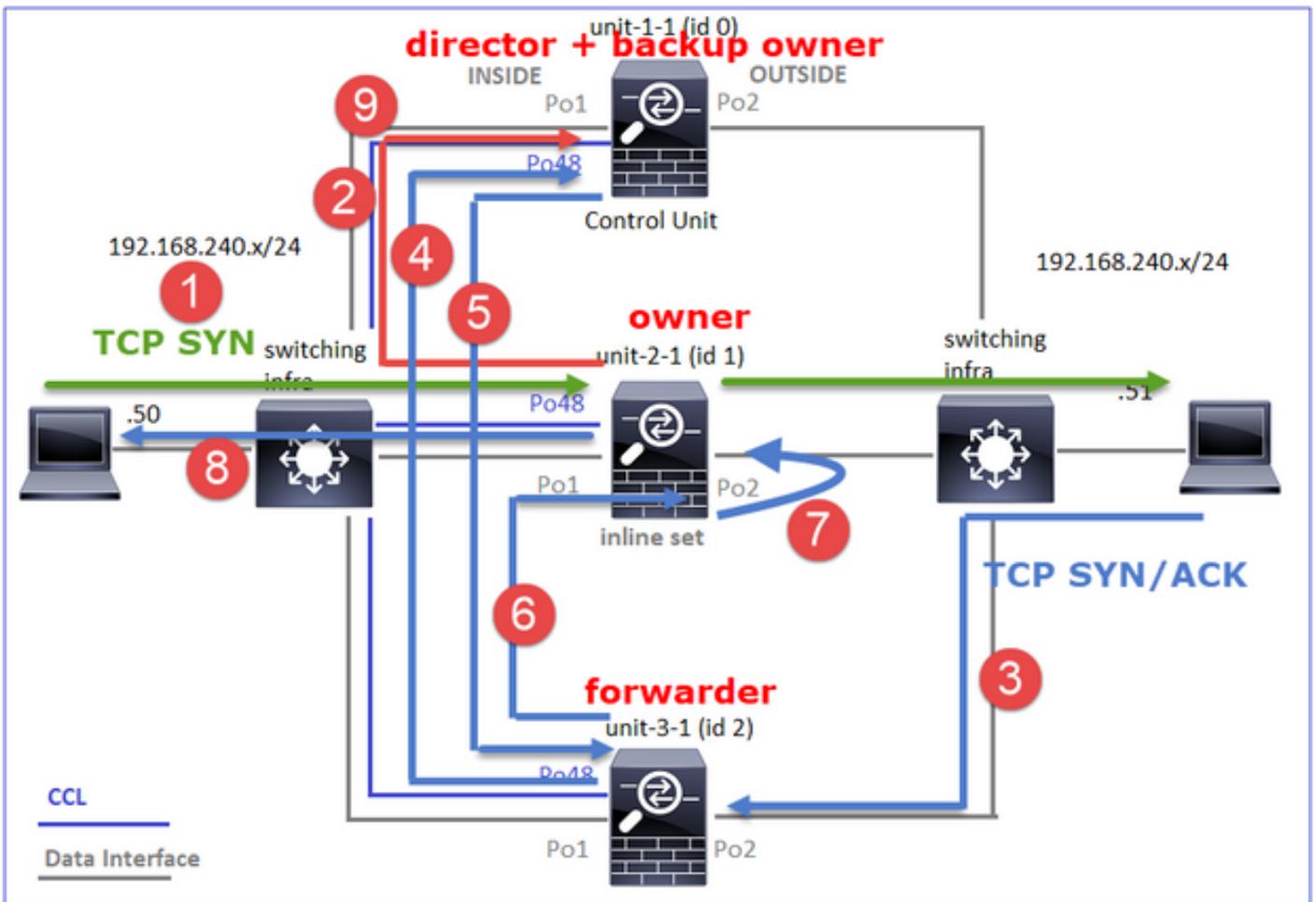
59210

, idle 0:00:00, bytes 0,

flags z

단위	플래그	메모
Unit-1-1	Y	· 디렉터 / 백업 소유자
Unit-2-1	b N	· Flow Owner(플로우 소유자) - 디바이스가 플로우를 처리합니다.
Unit-3-1	z	· 전달자

이는 다음과 같이 시각화할 수 있습니다.



1. TCP SYN 패킷이 Host-A에서 unit-2-1로 도착합니다. Unit-2-1이 플로우 소유자가 되고 unit-1-1이 디렉터로 선택됩니다
2. Unit-1-1은 백업 소유자(디렉터이므로)로 선택됩니다. 흐름 소유자가 UDP 4193에서 '클러스터 추가' 유니캐스트 메시지를 (으)로 전송합니다. 백업 소유자에게 플로우에 대해 알립니다.
3. TCP SYN/ACK 패킷이 Host-B에서 unit-3-1로 도착합니다. 흐름은 비대칭입니다.
4. Unit-3-1은 CCL을 통해 패킷을 디렉터(unit-1-1)로 전달한다.
5. Unit-1-1(디렉터)은 소유자가 unit-2-1임을 알고 패킷을 전달자(unit-3-1)로 다시 전송하여 소유자가 unit-2-1임을 알립니다.
6. Unit-3-1은 패킷을 unit-2-1(소유자)로 전송합니다.

7. Unit-2-1은 인터페이스 OUTSIDE의 패킷을 다시 삽입합니다.
8. Unit-2-1은 패킷을 Host-A로 전달합니다.
9. 연결이 종료되면 소유자는 클러스터 삭제 메시지를 보내 백업 소유자에서 플로우 정보를 제거합니다.

 참고: 2단계(CCL을 통한 패킷)가 4단계(데이터 트래픽)보다 먼저 발생하는 것이 중요합니다. 다른 경우(예를 들어, 경합 상태), 감독은 흐름을 인식하지 못한다. 따라서 인라인 집합이므로 패킷을 목적지로 전달합니다. 인터페이스가 인라인 집합에 없는 경우 데이터 패킷이 삭제됩니다.

관찰 3. 추적을 사용하여 캡처하면 비대칭 트래픽과 CCL을 통한 교환이 표시됩니다.

전달 트래픽(TCP SYN)

Unit-2-1(소유자)

<#root>

firepower#

```
cluster exec unit unit-2-1 show cap CAPI packet-number 1 trace
```

```
1: 09:19:49.760702 192.168.240.50.59210 > 192.168.240.51.80: S 4110299695:4110299695(0) win 29200 <mss>
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW
```

```
I (1) got initial, attempting ownership.
```

```
Phase: 2
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: NO FLOW
```

```
I (1) am becoming owner
```

반환 트래픽(TCP SYN/ACK)

Unit-3-1(ID 2 - 전달자)은 CCL을 통해 패킷을 unit-1-1(ID 0 - 디렉터)로 전송합니다.

<#root>

firepower#

cluster exec unit unit-3-1 show cap CAPO packet-number 1 trace

1: 09:19:49.760336 192.168.240.51.80 > 192.168.240.50.59210:

s

4209225081:4209225081(0)

ack

4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,wscale 7>

Phase: 1

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'OUTSIDE'

Flow type: NO FLOW

I (2) am asking director (0).

Unit-1-1(디렉터) - Unit-1-1(ID 0)은 플로우 소유자가 unit-2-1(ID 1)임을 알고 CCL을 통해 패킷을 다시 unit-3-1(ID 2 - 전달자)로 전송합니다.

<#root>

firepower#

cluster exec show cap CAPO packet-number 1 trace

unit-1-1(LOCAL):\*\*\*\*\*

1: 09:19:49.761038 192.168.240.51.80 > 192.168.240.50.59210:

s

4209225081:4209225081(0)

ack

4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,wscale 7>

Phase: 1

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'OUTSIDE'

Flow type: STUB

I (0) am director, valid owner (1), update sender (2).

Unit-3-1(ID 2 - 전달자)은 CCL을 통해 패킷을 가져와 unit-2-1(ID 1 - 소유자)로 전송합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-3-1 show cap CAPO packet-number 2 trace
```

```
...
```

```
2: 09:19:49.761008 192.168.240.51.80 > 192.168.240.50.59210:
```

```
s
```

```
4209225081:4209225081(0) ack 4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,w
```

```
Phase: 1
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'OUTSIDE'
```

```
Flow type: STUB
```

```
I (2) am becoming forwarder to (1), sender (0).
```

소유자는 패킷을 다시 거부하고 목적지로 전달합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec unit unit-2-1 show cap CAPO packet-number 2 trace
```

```
2: 09:19:49.775701 192.168.240.51.80 > 192.168.240.50.59210:
```

```
s
```

```
4209225081:4209225081(0)
```

```
ack
```

```
4110299696 win 28960 <mss 1460,sackOK,timestamp 567715984 130834570,nop,wscale 7>
```

```
Phase: 1
```

```
Type: CLUSTER-EVENT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Input interface: 'OUTSIDE'
```

```
Flow type: FULL
```

```
I (1) am owner, sender (2).
```

관찰 4. FTD 데이터 플레인 syslog는 모든 유닛에서 연결 생성 및 종료를 보여줍니다.

- Unit-1-1(디렉터/백업 소유자)
- Unit-2-1(소유자)
- Unit-3-1(전달자)

<#root>

firepower#

```
cluster exec show log | i 59210
```

unit-1-1(LOCAL):\*\*\*\*\*

Dec 03 2020 09:19:49: %FTD-6-302022:

Built director stub TCP connection

for INSIDE:192.168.240.50/59210 (192.168.240.50/59210) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)

Dec 03 2020 09:19:59: %FTD-6-302023:

Teardown director TCP connection

for INSIDE:192.168.240.50/59210 to OUTSIDE:192.168.240.51/80 duration 0:00:09 forwarded bytes 0 Cluste

unit-2-1:\*\*\*\*\*

Dec 03 2020 09:19:49: %FTD-6-302303:

Built TCP state-bypass connection

14483 from INSIDE:192.168.240.50/59210 (192.168.240.50/59210) to OUTSIDE:192.168.240.51/80 (192.168.240.51/80)

Dec 03 2020 09:19:59: %FTD-6-302304:

Teardown TCP state-bypass connection

14483 from INSIDE:192.168.240.50/59210 to OUTSIDE:192.168.240.51/80 duration 0:00:09 bytes 1024003336

unit-3-1:\*\*\*\*\*

Dec 03 2020 09:19:49: %FTD-6-302022:

Built forwarder stub TCP connection

for OUTSIDE:192.168.240.51/80 (192.168.240.51/80) to unknown:192.168.240.50/59210 (192.168.240.50/59210)

Dec 03 2020 09:19:59: %FTD-6-302023:

Teardown forwarder TCP connection

for OUTSIDE:192.168.240.51/80 to unknown:192.168.240.50/59210 duration 0:00:09 forwarded bytes 1024003

## 문제 해결

### 클러스터 문제 해결 소개

클러스터 문제는 다음과 같이 분류할 수 있습니다.

- 컨트롤 플레인 문제(클러스터 안정성과 관련된 문제)
- 데이터 플레인 문제(전송 트래픽과 관련된 문제)

## 클러스터 데이터 플레인 문제

### NAT/PAT 공통 문제

#### 중요한 구성 고려 사항

- PAT(Port Address Translation) 풀에는 최소한 클러스터의 유닛 수만큼 사용 가능한 IP가 있어야 하며, 클러스터 노드보다 더 많은 IP가 있어야 합니다.
- 기본 xlate per-session 명령은 비활성화할 특별한 이유가 없는 한 그대로 두어야 합니다. xlate per-session이 비활성화된 연결을 위해 작성된 모든 PAT xlate는 클러스터의 제어 노드 유닛에서 항상 처리되므로 성능이 저하될 수 있습니다.

클러스터 IP 불균형의 원인이 되는 로우 포트(low-port)에서 비롯된 트래픽으로 인해 높은 PAT 풀 범위 사용

FTD는 PAT IP를 범위로 분할하고 xlate를 동일한 소스 범위로 유지하려고 시도합니다. 이 표에서는 소스 포트가 동일한 소스 범위 내의 전역 포트로 변환되는 방법을 보여줍니다.

원래 소스 포트	변환된 소스 포트
1-511	1-511
512-1023	512-1023
1024-65535	1024-65535

소스 포트 범위가 짝 찢고 해당 범위에서 새 PAT xlate를 할당해야 하는 경우 FTD는 다음 IP로 이동하여 해당 소스 포트 범위에 대한 새 변환을 할당합니다.

#### 증상

클러스터를 지나는 NAT된 트래픽의 연결 문제

#### 확인

```
<#root>
```

```
#
```

```
show nat pool
```

FTD 데이터 플레인 로그에 PAT 풀 소진이 표시됩니다.

<#root>

```
Dec 9 09:00:00 192.0.2.10 FTD-FW %ASA-3-202010:
```

```
PAT pool exhausted. Unable to create TCP connection
```

```
from Inside:192.0.2.150/49464 to Outside:192.0.2.250/20015
```

```
Dec 9 09:00:00 192.0.2.10 FTD-FW %ASA-3-202010:
```

```
PAT pool exhausted. Unable to create TCP connection
```

```
from Inside:192.0.2.148/54141 to Outside:192.0.2.251/443
```

완화

NAT Flat Port Range(NAT 플랫폼 포트 범위)를 구성하고 Reserve Ports(예약 포트)를 포함합니다.

또한, post-6.7/9.15.1에서는 노드가 PAT가 적용되는 대규모 백그라운드 트래픽으로 클러스터를 나가거나 클러스터에 가입하는 경우에만 포트 블록 분배가 불균형하게 될 수 있습니다. 포트 블록이 해제되어 노드 간에 재배포되는 경우에만 복구됩니다.

포트 블록 기반 배포를 사용하는 경우 노드에 pb-1, pb-2 ... pb-10과 같은 10개의 포트 블록이 할당됩니다. 노드는 항상 첫 번째 사용 가능한 포트 블록에서 시작하여 만료될 때까지 임의의 포트를 할당합니다. 해당 지점까지의 모든 포트 블록이 모두 소진된 경우에만 할당이 다음 포트 블록으로 이동합니다.

예를 들어, 호스트가 512개의 연결을 설정하면 유닛에서는 pb-1의 모든 512개 연결에 대해 매핑된 포트를 임의로 할당합니다. 이제 이 512개의 연결이 모두 활성 상태일 때 pb-1이 모두 소진되어 호스트가 513번째 연결을 설정하면 pb-2로 이동하여 호스트에서 임의의 포트를 할당합니다. 이제 다시 513개의 연결 중에서 10번째 연결이 완료되고 pb-1에서 사용 가능한 하나의 포트가 지워졌다고 가정합니다. 이때 호스트가 514번째 연결을 설정하면 pb-1에 10번째 연결 제거의 일부로 릴리스된 자유 포트가 있으므로 pb-1에서 매핑된 포트가 할당되고 pb-2에서는 할당되지 않습니다.

가장 중요한 점은 여유 포트가 있는 첫 번째 가용 포트 블록에서 할당이 수행되므로, 정상적으로 로드된 시스템에서 마지막 포트 블록을 항상 재배포할 수 있습니다. 또한 PAT는 일반적으로 단기간 연결에 사용됩니다. 포트 블록이 더 짧은 시간에 이용 가능하게 될 확률은 매우 높다. 따라서 포트 블록 기반 풀 배포를 통해 풀 배포가 균형을 유지하는 데 필요한 시간을 개선할 수 있습니다.

그러나 pb-1에서 pb-10까지의 모든 포트 블록이 모두 소진되거나 각 포트 블록이 장기간 연결을 위해 포트를 보유하는 경우 포트 블록이 신속하게 해제되어 재배포되지 않습니다. 이러한 경우 가장 덜 파괴적인 접근 방식은 다음과 같습니다.

1. 과도한 포트 블록이 있는 노드를 식별합니다(show nat pool cluster summary).
2. 해당 노드에서 가장 적게 사용된 포트 블록을 식별합니다(show nat pool ip <addr> detail).
3. 재배포에 사용할 수 있도록 해당 포트 블록의 xlate를 지웁니다(xlate global <addr> gport 'start-end' 지우기).

---

 경고: 그러면 관련 연결이 중단됩니다.

---

다른 대상으로 리디렉션할 때 듀얼 채널 웹 사이트(예: 웹 메일, बैंकिंग 등) 또는 SSO 웹 사이트로 이

동할 수 없습니다.

## 증상

듀얼 채널 웹 사이트(예: 웹 메일, 은행 웹 사이트 등)를 찾아볼 수 없습니다. 사용자가 클라이언트가 두 번째 소켓/연결을 열도록 요구하는 웹 사이트에 연결하고 두 번째 연결이 첫 번째 연결을 해시된 것과 다른 클러스터 멤버에 해시되고 트래픽이 IP PAT 풀을 사용하는 경우, 서버가 다른 공용 IP 주소에서 연결을 수신할 때 트래픽이 재설정됩니다.

## 확인

데이터 플레인 클러스터 캡처를 통해 영향을 받는 전송 흐름이 어떻게 처리되는지 확인합니다. 이 경우 TCP 재설정은 대상 웹 사이트에서 제공됩니다.

## 완화(6.7 이전/9.15.1)

- 다중 세션 애플리케이션이 여러 매핑된 IP 주소를 사용하는지 확인합니다.
- `show nat pool cluster summary` 명령을 사용하여 풀이 균등하게 배포되었는지 확인합니다.
- `cluster exec show conn` 명령을 사용하여 트래픽이 올바르게 로드 밸런싱되는지 확인합니다.
- `show nat pool cluster ip <address> detail` 명령을 사용하여 스티커 IP의 풀 사용량을 확인합니다.
- 스티커 IP를 사용하지 못한 연결을 확인하려면 `syslog 305021(6.7/9.15)`를 활성화합니다.
- 문제를 해결하려면 PAT 풀에 IP를 더 추가하거나 연결된 스위치에서 로드 밸런싱 알고리즘을 세부적으로 조정합니다.

## 이더 채널 로드 밸런싱 알고리즘 정보:

- 비 FP9300의 경우 및 한 서버에서 인증이 발생하는 경우: 인접한 스위치의 이더 채널 로드 밸런싱 알고리즘을 Source IP/Port 및 Destination IP/Port에서 Source IP 및 Destination IP로 조정합니다.
- 비 FP9300 및 여러 서버를 통해 인증이 발생하는 경우: 인접한 스위치의 이더 채널 로드 밸런싱 알고리즘을 Source IP/Port 및 Destination IP/Port에서 Source IP로 조정합니다.
- FP9300의 경우 FP9300 샤페에서 로드 밸런싱 알고리즘은 `source-dest-port source-dest-ip source-dest-mac`으로 고정되어 있으므로 변경할 수 없습니다. 이 경우 해결 방법은 FlexConfig를 사용하여 FTD 컨피그레이션에 `xlate per-session deny` 명령을 추가하여 특정 대상 IP 주소(문제가 있거나 호환되지 않는 애플리케이션의 경우)에 대한 트래픽을 인트라 샤페 클러스터의 제어 노드에서만 처리하도록 하는 것입니다. 해결 방법은 다음과 같은 부작용과 함께 제공됩니다.
  - 다르게 변환된 트래픽의 로드 밸런싱이 없습니다(모든 것이 제어 노드로 이동됨).
  - `xlate` 슬롯이 소진될 수 있으며 제어 노드의 다른 트래픽에 대한 NAT 변환에 악영향을 미칠 수 있습니다.
  - 샤페 내 클러스터의 확장성 감소.

풀에 충분한 PAT IP가 없기 때문에 제어 노드로 전송된 모든 트래픽으로 인해 클러스터 성능이 저하됩니다.

## 증상

클러스터에 PAT IP가 부족하여 데이터 노드에 사용 가능한 IP를 할당할 수 없습니다. 따라서 PAT

컨피그레이션의 대상이 되는 모든 트래픽이 처리를 위해 제어 노드로 전달됩니다.

## 확인

show nat pool cluster 명령을 사용하여 각 유닛에 대한 할당을 보고 모두 풀에 하나 이상의 IP를 소유하고 있는지 확인합니다.

## 완화

6.7/9.15.1 이전의 경우 클러스터의 노드 수와 최소 같은 크기의 PAT 풀이 있어야 합니다. PAT 풀이 있는 6.7/9.15.1 이후에서는 모든 PAT 풀 IP에서 포트 블록을 할당합니다. PAT 풀 사용량이 너무 많아 풀이 자주 소진되는 경우 PAT 풀 크기를 늘려야 합니다(FAQ 섹션 참조).

xlate가 세션당 활성화되지 않았기 때문에 제어 노드로 전송된 모든 트래픽으로 인해 성능이 저하됩니다.

## 증상

많은 고속 UDP 백업 흐름이 클러스터 제어 노드를 통해 처리되므로 성능에 영향을 줄 수 있습니다.

## 배경

PAT를 사용하는 데이터 노드에서는 세션당 활성화된 xlate를 사용하는 연결만 처리할 수 있습니다. show run all xlate 명령을 사용하여 xlate per-session 컨피그레이션을 확인합니다.

Per-Session Enabled는 연결된 연결이 끊어질 때 xlate가 즉시 해제됨을 의미합니다. 이렇게 하면 연결이 PAT를 받을 때 초당 연결 성능이 향상됩니다. 비 세션당 xlate는 연결된 연결이 끊긴 후 30초 더 기다립니다. 연결 속도가 충분히 높으면 각 글로벌 IP에서 사용 가능한 65k TCP/UDP 포트를 단시간에 모두 사용할 수 있습니다.

기본적으로 모든 TCP 트래픽은 xlate당 활성화되며 UDP DNS 트래픽만 세션당 활성화됩니다. 즉, 모든 비 DNS UDP 트래픽이 처리를 위해 제어 노드로 전달됩니다.

## 확인

클러스터 유닛 간의 연결 및 패킷 배포를 확인하려면 다음 명령을 사용합니다.

```
<#root>
```

```
firepower#
```

```
show cluster info conn-distribution
```

```
firepower#
```

```
show cluster info packet-distribution
```

```
firepower#
```

```
show cluster info load-monitor
```

UDP 연결을 소유하는 클러스터 노드를 확인하려면 `cluster exec show conn` 명령을 사용합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show conn
```

이 명령을 사용하여 클러스터 노드 간 풀 사용을 이해할 수 있습니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show nat pool ip
```

| in UDP

## 완화

관심 트래픽(예: UDP)에 대한 세션당 PAT(세션당 `permit udp` 명령)를 구성합니다. ICMP의 경우 기본 다중 세션 PAT에서 변경할 수 없으므로 PAT가 구성된 경우 ICMP 트래픽은 제어 노드에서 항상 처리됩니다.

노드가 클러스터를 나가거나 조인할 때 PAT 풀 분배가 불균형해집니다.

## 증상

- PAT IP 할당은 클러스터에서 나가거나 클러스터에 참가하는 유닛으로 인해 시간이 지남에 따라 불균형해질 수 있으므로 연결 문제가 발생합니다.
- post-6.7/9.15.1에서는 새로 연결된 노드가 충분한 포트 블록을 가져올 수 없는 경우가 있습니다. 포트 블록이 없는 노드는 제어 노드로 트래픽을 리디렉션합니다. 하나 이상의 포트 블록이 있는 노드는 트래픽을 처리하고 풀이 소진되면 삭제합니다.

## 확인

- 데이터 플레인 syslog에는 다음과 같은 메시지가 표시됩니다.

<#root>

%ASA-3-202010:

```
NAT pool exhausted. Unable to create TCP connection
from inside:192.0.2.1/2239 to outside:192.0.2.150/80
```

- show nat pool cluster summary 명령을 사용하여 풀 배포를 식별할 수 있습니다.
- cluster exec show nat pool ip <addr> detail 명령을 사용하여 클러스터 노드 간 풀 사용량을 파악합니다.

완화

- 6.7/9.15.1 이전의 몇 가지 해결 방법은 Cisco 버그 ID CSCvd에 설명되어 [있습니다10530](#) 
- post-6.7/9.15.1에서 clear xlate global <ip> gport <start-end> 명령을 사용하여 필요한 노드로 재배포하기 위해 다른 노드의 일부 포트 블록을 수동으로 지웁니다.

증상

클러스터에서 PAT하는 트래픽의 주요 연결 문제. 이는 FTD 데이터 플레인에서 설계별로 전역 NAT 주소에 대해 GARP를 전송하지 않기 때문입니다.

확인

직접 연결된 디바이스의 ARP 테이블에는 제어 노드 변경 후 클러스터 데이터 인터페이스의 MAC 주소가 다르게 표시됩니다.

<#root>

```
root@kali2:~/tests#
```

```
arp -a
```

```
? (192.168.240.1) at f4:db:e6:
```

```
33:44:2e
```

```
[ether] on eth0
```

```
root@kali2:~/tests#
```

```
arp -a
```

```
? (192.168.240.1) at f4:db:e6:
```

```
9e:3d:0e
```

```
[ether] on eth0
```

완화

클러스터 데이터 인터페이스에 정적(가상) MAC을 구성합니다.

PAT가 실패한 연결

증상

클러스터에서 PAT하는 트래픽에 대한 연결 문제.

확인/완화

- 컨피그레이션이 제대로 복제되었는지 확인합니다.
- 풀이 균등하게 분포되어 있는지 확인합니다.
- 풀 소유권이 유효한지 확인하십시오.
- show asp cluster 카운터에 오류 카운터 증분이 없습니다.
- 디렉터/전달자 플로우가 적절한 정보로 생성되었는지 확인합니다.
- 백업 xlate가 작성, 업데이트 및 예상대로 정리되었는지 확인합니다.
- "세션당" 동작에 따라 xlate가 생성 및 종료되는지 확인합니다.
- 오류를 표시하려면 "debug nat 2"를 활성화합니다. 이 출력은 다음과 같이 매우 시끄러울 수 있습니다.

<#root>

firepower#

debug nat 2

nat:

no free blocks available to reserve for 192.168.241.59, proto 17

nat: no free blocks available to reserve for 192.168.241.59, proto 17

nat: no free blocks available to reserve for 192.168.241.58, proto 17

nat: no free blocks available to reserve for 192.168.241.58, proto 17

nat: no free blocks available to reserve for 192.168.241.57, proto 17

디버그를 중지하려면

<#root>

firepower#

un all

- 연결 및 NAT 관련 syslog를 활성화하여 정보를 실패한 연결과 연계합니다.

ASA 및 FTD 클러스터링 PAT 개선(9.15 이후 및 6.7)

무엇이 바뀌었습니까?

PAT 작업이 다시 설계되었습니다. 개별 IP는 더 이상 각 클러스터 멤버에 배포되지 않습니다. 그 대신 PAT IP는 포트 블록으로 분할되고 IP 고착성 작업과 함께 클러스터 멤버 간에 균등하게(가능한 한 많이) 분배됩니다.

새 설계에서는 다음과 같은 제한을 해결합니다(이전 섹션 참조).

- 클러스터 전반의 IP 고착성이 부족하기 때문에 다중 세션 애플리케이션이 영향을 받습니다.
- 최소 클러스터의 노드 수와 같은 크기의 PAT 풀이 필요합니다.
- 노드가 클러스터를 나가거나 조인할 때 PAT 풀 분배가 불균형해집니다.
- PAT 풀 불균형을 나타내는 syslog가 없습니다.

기술적으로 기본 1-511, 512-1023 및 1024-65535 포트 범위 대신, 이제 PAT의 기본 포트 범위로 1024-65535이 있습니다. 이 기본 범위는 일반 PAT에 대한 특권 포트 범위 1~1023을 포함하도록 확장할 수 있습니다('include-reserve' 옵션).

FTD 6.7의 PAT 풀 컨피그레이션의 예입니다. 자세한 내용은 컨피그레이션 설명서의 관련 섹션을 참조하십시오.

**NAT Rule:**  
Manual NAT Rule

**Insert:**  
In Category NAT Rules Before

**Type:**  
Dynamic

Enable

Description:

---

Interface Objects   **Translation**   PAT Pool   Advanced

Original Packet	Translated Packet
Original Source:* net_192.168.240.0 +	Translated Source: Address
Original Destination: Address +	+ +
Original Source Port: + +	Translated Destination: + +
Original Destination Port: + +	Translated Source Port: + +
	Translated Destination Port: + +

Interface Objects Translation **PAT Pool** Advanced

Enable PAT Pool

PAT:

Address  +

Use Round Robin Allocation

Extended PAT Table

Flat Port Range ⓘ This option always enabled on device from v6.7.0 irrespective of its configured value.

Include Reserve Ports

Block Allocation

## PAT에 대한 추가 트러블슈팅 정보

FTD 데이터 플레인 syslog(post-6.7/9.15.1)

클러스터 노드의 고정 IP에서 모든 포트가 소진되고 할당이 사용 가능한 다음 IP로 이동하는 경우 고정 무효화 syslog가 생성됩니다. 예를 들면 다음과 같습니다.

```
%ASA-4-305021: Ports exhausted in pre-allocated PAT pool IP 192.0.2.100 for host 198.51.100.100 Allocat
```

풀 불균형 syslog는 클러스터에 참가할 때 노드에 생성되며 다음과 같이 포트 블록의 일부 또는 동일하지 않은 공유를 가져오지 않습니다.

```
%ASA-4-305022: Cluster unit ASA-4 has been allocated 0 port blocks for PAT usage. All units should have
```

```
%ASA-4-305022: Cluster unit ASA-4 has been allocated 12 port blocks for PAT usage. All units should have
```

## Show 명령

### 풀 배포 상태

show nat pool cluster summary 출력에서 각 PAT IP 주소에 대해 균형 잡힌 배포 시나리오에서 노드 전체에서 1개 이상의 포트 블록 차이가 없어야 합니다. 균형 잡힌 불균형 포트 블록 분배의 예.

```
<#root>
```

```
firepower#
```

```
show nat pool cluster summary
```

```
port-blocks count display order: total, unit-1-1, unit-2-1, unit-3-1
IP OUTSIDE:ip_192.168.241.57-59 192.168.241.57 (126 -
```

42 / 42 / 42

```
)  
IP OUTSIDE:ip_192.168.241.57-59 192.168.241.58 (126 - 42 / 42 / 42)  
IP OUTSIDE:ip_192.168.241.57-59 192.168.241.59 (126 - 42 / 42 / 42)
```

분배 불균형:

```
<#root>
```

```
firepower#
```

```
show nat pool cluster summary
```

```
port-blocks count display order: total, unit-1-1, unit-4-1, unit-2-1, unit-3-1  
IP outside:src_map 192.0.2.100 (128 - 32 /
```

```
22 / 38
```

```
/ 36)
```

풀 소유권 상태

show nat pool cluster 출력에는 소유자 또는 UNKNOWN으로 백업하는 단일 포트 블록이 없어야 합니다. 하나가 있으면 풀 소유권 통신에 문제가 있음을 나타냅니다. 예:

```
<#root>
```

```
firepower#
```

```
show nat pool cluster | in
```

```
[3072-3583], owner unit-4-1, backup <
```

```
UNKNOWN
```

```
>
```

```
[56832-57343], owner <UNKNOWN>, backup <UNKNOWN>
```

```
[10240-10751], owner unit-2-1, backup <UNKNOWN>
```

포트 블록에서 포트 할당 어카운팅

show nat pool 명령은 추가 옵션으로 개선되어 자세한 정보와 필터링된 출력을 표시합니다. 예:

```
<#root>
```

```
firepower#
```

```
show nat pool detail
```

```

TCP PAT pool INSIDE, address 192.168.240.1, range 1-1023, allocated 0
TCP PAT pool INSIDE, address 192.168.240.1, range 1024-65535, allocated 18
UDP PAT pool INSIDE, address 192.168.240.1, range 1-1023, allocated 0
UDP PAT pool INSIDE, address 192.168.240.1, range 1024-65535, allocated 20
TCP PAT pool OUTSIDE, address 192.168.241.1, range 1-1023, allocated 0
TCP PAT pool OUTSIDE, address 192.168.241.1, range 1024-65535, allocated 18
UDP PAT pool OUTSIDE, address 192.168.241.1, range 1-1023, allocated 0
UDP PAT pool OUTSIDE, address 192.168.241.1, range 1024-65535, allocated 20
UDP PAT pool OUTSIDE, address 192.168.241.58
range 1024-1535, allocated 512
range 1536-2047, allocated 512
range 2048-2559, allocated 512
range 2560-3071, allocated 512
...
unit-2-1:*****
UDP PAT pool OUTSIDE, address 192.168.241.57
range 1024-1535, allocated 512 *
range 1536-2047, allocated 512 *
range 2048-2559, allocated 512 *

```

\*'는 백업된 포트 블록임을 나타냅니다.

이 문제를 해결하려면 `clear xlate global <ip> gport <start-end>` 명령을 사용하여 필요한 노드로 재 배포하기 위해 다른 노드의 일부 포트 블록을 수동으로 지웁니다.

수동으로 트리거된 포트 블록 재배포

- 지속적인 트래픽이 있는 생산 네트워크에서 노드가 클러스터를 나갔다가 다시 연결할 때(역 추적 때문일 수 있음), 동일한 풀 점유율을 가져올 수 없거나 최악의 경우 포트 블록을 가져올 수 없는 경우가 있습니다.
- `show nat pool cluster summary` 명령을 사용하여 어떤 노드가 필요 이상의 포트 블록을 소유 하는지 식별할 수 있습니다.
- 더 많은 포트 블록을 소유하는 노드에서 `show nat pool ip <addr> detail` 명령을 사용하여 할당 수가 가장 적은 포트 블록을 확인합니다.
- `clear xlate global <address> gport <start-end>` 명령을 사용하여 이러한 포트 블록에서 생성 된 변환을 지워서 필수 노드로 재배포할 수 있습니다. 예를 들면 다음과 같습니다.

```
<#root>
```

```
firepower#
```

```
show nat pool detail | i 19968
```

```

range 19968-20479, allocated 512
range 19968-20479, allocated 512
range 19968-20479, allocated 512

```

```
firepower#
```

```
clear xlate global 192.168.241.57 gport 19968-20479
```

## 포스트 6.7/9.15.1 PAT 자주 묻는 질문(FAQ)

Q. 클러스터에서 사용 가능한 유닛 수에 사용 가능한 IP 수가 있는 경우에도 유닛당 IP 1개를 옵션으로 사용할 수 있습니까?

A. 더 이상 그렇지 않으며, IP 주소 기반 풀 분배 체계와 포트 블록 기반 풀 분배 체계 사이를 전환하기 위한 토글(toggle)이 없습니다.

IP 주소 기반 풀 배포의 이전 체계는 호스트의 여러 연결(단일 애플리케이션 트랜잭션의 일부)이 클러스터의 여러 노드로 로드 밸런싱되는 다중 세션 애플리케이션 오류를 발생시켰으며, 이에 따라 서로 다른 매핑된 IP 주소로 변환되어 대상 서버가 다른 엔티티에서 소싱한 것으로 표시됩니다.

또한 새 포트 블록 기반 배포 구성에서는 단일 PAT IP 주소만큼 낮게 작업할 수 있지만 항상 PAT가 필요한 연결 수를 기준으로 충분한 PAT IP 주소를 사용하는 것이 좋습니다.

Q. 클러스터에 대한 PAT 풀의 IP 주소 풀이 남아 있습니까?

A. 네, 가능합니다. 모든 PAT 풀 IP의 포트 블록은 클러스터 노드 전체에 분산됩니다.

Q. PAT 풀에 대해 여러 IP 주소를 사용하는 경우, 각 IP 주소당 각 멤버에 제공되는 동일한 포트 블록입니까?

A. 아니요. 각 IP는 독립적으로 배포됩니다.

Q. 모든 클러스터 노드에는 모든 공용 IP가 있지만 포트의 하위 집합만 있습니까? 그렇다면 소스 IP가 동일한 공용 IP를 사용할 때마다 보장되는 것일까요?

A. 정확합니다. 각 PAT IP는 각 노드에서 부분적으로 소유합니다. 선택한 공용 IP가 노드에서 소진되면 스티커 IP를 보존할 수 없음을 나타내는 syslog가 생성되고 할당이 사용 가능한 다음 공용 IP로 이동합니다. 독립형, HA 또는 클러스터 구축의 경우, IP 고착성은 항상 풀 가용성에 따라 최선의 노력을 기울입니다.

Q. 모든 것이 PAT 풀의 단일 IP 주소를 기반으로 합니까? 그러나 PAT 풀의 IP 주소가 두 개 이상 사용되는 경우에는 적용되지 않습니까?

A. PAT 풀의 여러 IP 주소에도 적용됩니다. PAT 풀에 있는 모든 IP의 포트 블록은 클러스터 노드 전체에 분산됩니다. PAT 풀의 모든 IP 주소는 클러스터의 모든 멤버에서 분할됩니다. 따라서 PAT 풀에 주소의 클래스 C가 있는 경우 모든 클러스터 멤버에는 각 PAT 풀 주소의 포트 풀이 있습니다.

Q. CGNAT와 연동됩니까?

A. 예. CGNAT도 지원됩니다. 블록 할당 PAT라고도 하는 CGNAT의 기본 블록 크기는 '512'이며 xlate 블록 할당 크기 CLI를 통해 수정할 수 있습니다. 일반 동적 PAT(non-CGNAT)의 경우, 블록 크기는 항상 '512'로 고정되어 구성할 수 없습니다.

Q. 유닛이 클러스터를 벗어나면 제어 노드는 포트 블록 범위를 다른 유닛에 할당하거나 그대로 유

지합니까?

A. 각 포트 블록에는 소유자 및 백업이 있습니다. xlate는 포트 블록에서 생성될 때마다 포트 블록 백업 노드에도 복제됩니다. 노드가 클러스터를 떠나면 백업 노드는 모든 포트 블록 및 모든 현재 연결을 소유합니다. 백업 노드는 이러한 추가 포트 블록의 소유자가 되었으므로 해당 노드에 대해 새 백업을 선택하고 모든 현재 xlate를 해당 노드에 복제하여 오류 시나리오를 처리합니다.

Q. 해당 경고를 기준으로 고착성을 적용하기 위해 취할 수 있는 조치는 무엇입니까?

A. 끈적임을 유지할 수 없는 이유는 두 가지다.

이유-1: 노드 중 하나에서 다른 노드보다 더 많은 수의 연결을 확인하므로 트래픽의 로드 밸런싱이 잘못되어 특정 스티커 IP 소진이 발생합니다. 트래픽이 클러스터 노드 전체에 고르게 분산되도록 하는 경우 이 문제를 해결할 수 있습니다. 예를 들어, FPR41xx 클러스터에서 연결된 스위치의 로드 밸런싱 알고리즘을 조정합니다. FPR9300 클러스터에서 새시 전체의 블레이드 수가 동일한지 확인합니다.

이유-2: PAT 풀 사용량이 매우 많아 풀이 자주 소모됩니다. 이를 해결하려면 PAT 풀 크기를 늘립니다.

Q. extended 키워드에 대한 지원은 어떻게 처리됩니까? 이 명령은 오류를 표시하고 업그레이드 중에 전체 NAT 명령이 추가되지 않게 합니다, 아니면 extended 키워드를 제거하고 경고를 표시합니까?

A. PAT 확장 옵션은 ASA 9.15.1/FP 6.7 이상의 클러스터에서 지원되지 않습니다. 컨피그레이션 옵션은 CLI/ASDM/CSM/FMC에서 제거되지 않습니다. 업그레이드를 통해 직접 또는 간접적으로 구성할 경우 경고 메시지가 표시되고, 컨피그레이션이 승인되지만 PAT의 확장된 기능이 작동한다는 것을 볼 수 없습니다.

Q. 동시 접속과 동일한 수의 번역입니까?

A. Pre-6.7/9.15.1에서는 1-65535이었지만, 소스 포트가 1-1024 범위에서 많이 사용되지 않으므로 1024-65535(64512 conn)이 됩니다. Post-6.7/9.15.1 구현에서는 'flat'을 기본 동작으로 사용하며 1024-65535입니다. 그러나 1-1024를 사용하려면 "include-reserve" 옵션을 사용할 수 있습니다.

Q. 노드가 클러스터에 다시 조인하는 경우 이전 백업 노드가 백업으로 사용되고 해당 백업 노드는 이전 포트 블록을 제공합니까?

A. 해당 시점의 포트 블록 가용성에 따라 달라집니다. 노드가 클러스터를 벗어나면 해당 포트 블록이 모두 백업 노드로 이동됩니다. 그런 다음 자유 포트 블록을 축적하고 필수 노드에 배포하는 제어 노드입니다.

Q. 제어 노드의 상태가 변경되면 새 제어 노드를 선택합니까, 아니면 PAT 블록 할당이 유지됩니까, 아니면 새 제어 노드를 기준으로 포트 블록이 재할당됩니까?

A. 새 제어 노드는 어떤 블록이 할당되었는지, 어떤 블록이 무료이며 어떤 블록에서 시작되는지를 파악합니다.

Q. xlate의 최대 수는 이 새 동작을 사용하는 최대 동시 연결 수와 같습니까?

A. 예. 최대 xlate 수는 PAT 포트의 가용성에 따라 달라집니다. 최대 동시 연결 수와는 상관이 없습니다. 주소 1개만 허용할 경우 가능한 연결 65535 있습니다. 더 필요한 경우 더 많은 IP 주소를 할당해야 합니다. 충분한 주소/포트가 있는 경우 최대 동시 연결에 도달할 수 있습니다.

Q. 새 클러스터 멤버가 추가될 때 포트 블록 할당의 프로세스는 무엇입니까? 재부팅 때문에 클러스터 멤버가 추가되면 어떻게 됩니까?

A. 포트 블록은 항상 제어 노드에 의해 분배됩니다. 포트 블록은 사용 가능한 포트 블록이 있는 경우에만 새 노드에 할당됩니다. 사용 가능한 포트 블록은 포트 블록 내의 매핑된 포트를 통해 어떤 연결도 제공되지 않음을 의미합니다.

또한 다시 조인할 때 각 노드는 소유할 수 있는 블록 수를 다시 계산합니다. 노드가 예정보다 많은 블록을 보유할 경우, 제어 노드가 사용 가능해질 때 추가 포트 블록을 릴리스합니다. 그런 다음 제어 노드는 새로 조인된 데이터 노드에 할당합니다.

Q. TCP 및 UDP 프로토콜 또는 SCTP만 지원됩니까?

A. SCTP는 동적 PAT에서 지원되지 않았습니다. SCTP 트래픽의 경우 고정 네트워크 객체 NAT만 사용하는 것이 좋습니다.

Q. 노드에 블록 포트가 부족하면 패킷을 삭제하고 사용 가능한 다음 IP 블록을 사용하지 않습니까?

A. 아니요, 바로 떨어지지 않습니다. 다음 PAT IP에서 사용 가능한 포트 블록을 사용합니다. 모든 PAT IP의 모든 포트 블록이 모두 소진되면 트래픽이 삭제됩니다.

Q. 클러스터 업그레이드 창에서 제어 노드의 오버로드를 방지하려면 제어 노드에서 모든 연결이 처리될 때까지 기다리는 것보다 이전에 수동으로 새 제어(예: 4유닛 클러스터 업그레이드의 중간 단계)를 선택하는 것이 더 나은가요?

A. 컨트롤을 마지막으로 업데이트해야 합니다. 이는 제어 노드가 최신 버전을 실행할 때 모든 노드가 최신 버전을 실행하지 않는 한 폴 배포를 시작하지 않기 때문입니다. 또한 업그레이드가 실행되면 최신 버전의 모든 데이터 노드가 이전 버전을 실행하는 경우 제어 노드의 폴 배포 메시지를 무시합니다.

이를 자세히 설명하려면 4개의 노드 A, B, C, D를 A와 함께 제어로 사용하는 클러스터 구축을 고려하십시오. 일반적인 무중단 업그레이드 단계는 다음과 같습니다.

1. 각 노드에 새 버전을 다운로드합니다.
2. 유닛 'D'를 다시 로드합니다. 모든 연결, xlate가 백업 노드로 이동합니다.
3. 유닛 'D'가 나타나고:

a. PAT 컨피그레이션 처리

b. 각 PAT IP를 포트 블록으로 분할

c. 모든 포트 블록이 할당되지 않은 상태입니다.

d. 컨트롤에서 받은 클러스터 PAT 메시지의 이전 버전을 무시합니다.

e. 모든 PAT 연결을 기본으로 리디렉션합니다.

4. 마찬가지로 새 버전의 다른 노드를 표시합니다.
5. A유닛의 컨트롤을 다시 로드합니다. 제어를 위한 백업이 없으므로 모든 기존 연결이 삭제됩니다
6. 새 컨트롤에서 포트 블록의 배포를 새 형식으로 시작합니다
7. 'A' 유닛이 다시 합류하여 포트 블록 분배 메시지를 수락하고 처리할 수 있습니다.

## 프래그먼트 처리

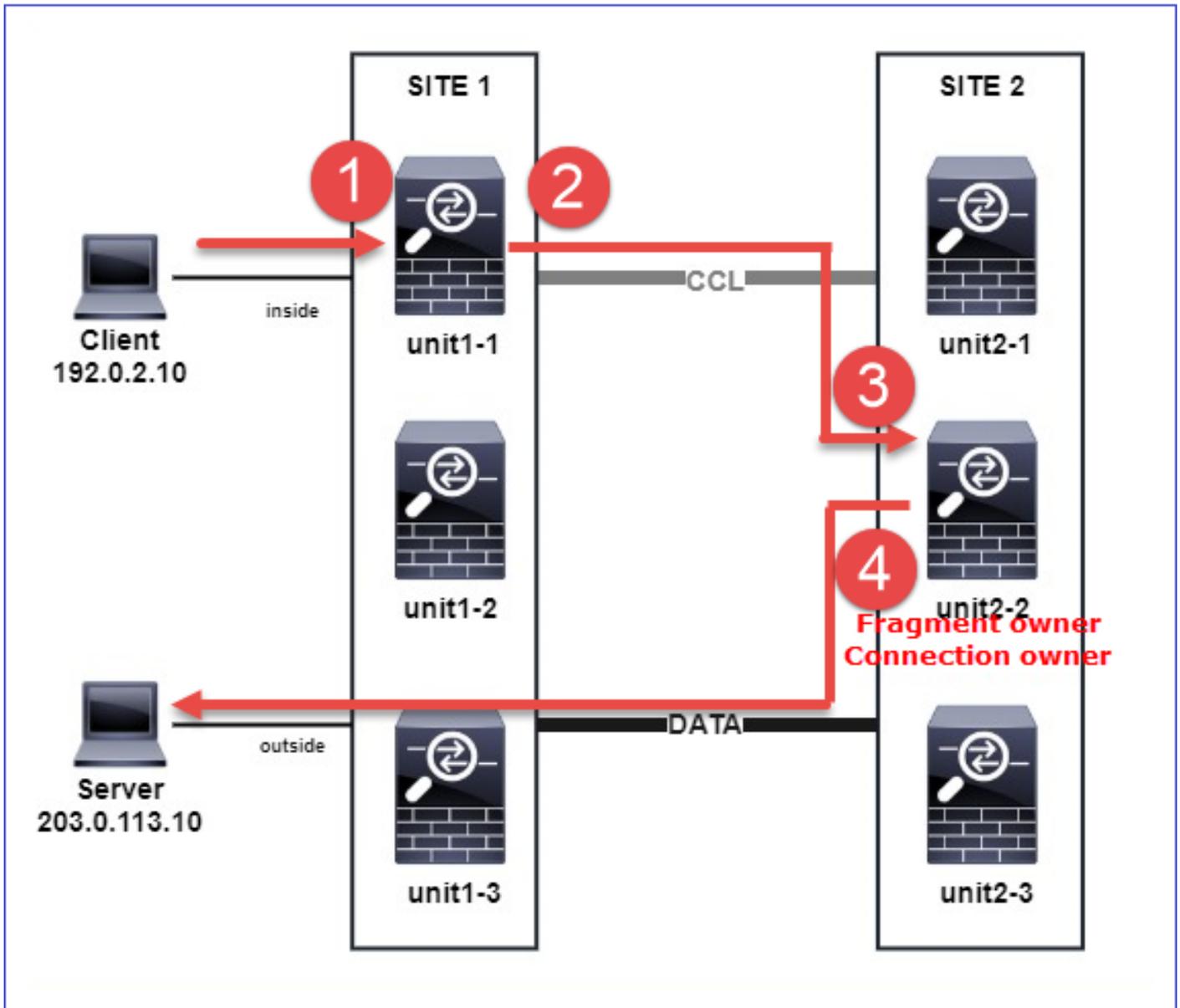
### 증상

사이트 간 클러스터 구축에서는 하나의 특정 사이트(사이트-로컬 트래픽)에서 처리해야 하는 프래그먼트된 패킷을 다른 사이트의 유닛으로 전송할 수 있습니다. 이러한 사이트 중 하나에는 프래그먼트 소유자가 있을 수 있기 때문입니다.

클러스터 논리에서는 프래그먼트된 패킷이 있는 연결에 대해 추가 역할이 정의됩니다. 조각 소유자

프래그먼트화된 패킷의 경우 프래그먼트를 수신하는 클러스터 유닛은 프래그먼트 소스 IP 주소, 목적지 IP 주소, 패킷 ID의 해시를 기반으로 프래그먼트 소유자를 결정합니다. 그런 다음 모든 프래그먼트는 클러스터 제어 링크를 통해 프래그먼트 소유자에게 전달됩니다. 첫 번째 프래그먼트에만 스위치 부하 균형 해시에 사용되는 5-튜플이 포함되므로 프래그먼트는 다른 클러스터 유닛으로 부하 균형을 이룰 수 있습니다. 다른 프래그먼트는 소스 및 대상 포트를 포함하지 않으며 다른 클러스터 유닛으로 로드 밸런싱될 수 있습니다. 프래그먼트 소유자는 소스/목적지 IP 주소 및 포트의 해시를 기반으로 디렉터를 결정할 수 있도록 패킷을 일시적으로 리어셈블합니다. 새 연결인 경우 조각 소유자가 연결 소유자가 됩니다. 기존 연결인 경우 프래그먼트 소유자는 모든 프래그먼트를 클러스터 제어 링크를 통해 연결 소유자에게 전달합니다. 그러면 연결 소유자가 모든 조각을 다시 어셈블합니다.

클라이언트에서 서버로의 조각화된 ICMP 에코 요청의 흐름과 함께 이 토폴로지를 고려하십시오.



작업의 순서를 이해하기 위해 trace 옵션으로 구성된 내부, 외부 및 클러스터 제어 링크 인터페이스에 클러스터 전체 패킷 캡처가 있습니다. 또한 inside 인터페이스에 reinject-hide 옵션으로 패킷 캡처가 구성됩니다.

```
<#root>
```

```
firepower#
```

```
cluster exec capture capi interface inside trace match icmp any any
```

```
firepower#
```

```
cluster exec capture capir interface inside reinject-hide trace match icmp any any
```

```
firepower#
```

```
cluster exec capture capo interface outside trace match icmp any any
```

```
firepower#
```

```
cluster exec capture capccl interface cluster trace match icmp any any
```

클러스터 내의 작업 순서:

1. 사이트 1의 unit-1-1은 프래그먼트된 ICMP 에코 요청 패킷을 수신합니다.

```
<#root>
```

```
firepower#
```

```
cluster exec show cap capir
```

```
unit-1-1(LOCAL)
```

```
:*****
```

```
2 packets captured
```

```
1: 20:13:58.227801 802.1Q vlan#10 P0 192.0.2.10 > 203.0.113.10 icmp: echo request
```

```
2: 20:13:58.227832 802.1Q vlan#10 P0
```

```
2 packets shown
```

2. unit-1-1은 사이트 2에서 유닛-2-2를 프래그먼트 소유자로 선택하고 프래그먼트화된 패킷을 사이트 2에 전송합니다.

유닛-1-1에서 유닛-2-2로 전송된 패킷의 목적지 MAC 주소는 유닛-2-2의 CCL 링크의 MAC 주소입니다.

```
<#root>
```

```
firepower#
```

```
show cap capccl packet-number 1 detail
```

```
7 packets captured
```

```
1: 20:13:58.227817
```

```
0015.c500.018f 0015.c500.029f
```

```
0x0800 Length: 1509
```

```
192.0.2.10 > 203.0.113.10
```

```
icmp: echo request (wrong icmp csum) (frag 46772:1475@0+) (ttl 3)
```

1 packet shown

firepower#

show cap capccl packet-number 2 detail

7 packets captured

2: 20:13:58.227832

0015.c500.018f 0015.c500.029f

0x0800 Length: 637

192.0.2.10 > 203.0.113.10

(

frag 46772

:603@1480) (ttl 3)

1 packet shown

firepower#

cluster exec show interface po48 | i MAC

unit-1-1(LOCAL):\*\*\*\*\*

MAC address 0015.c500.018f, MTU 1500

unit-1-2:\*\*\*\*\*

MAC address 0015.c500.019f, MTU 1500

unit-2-2

:\*\*\*\*\*

MAC address 0015.c500.029f, MTU 1500

unit-1-3:\*\*\*\*\*

MAC address 0015.c500.016f, MTU 1500

unit-2-1:\*\*\*\*\*

MAC address 0015.c500.028f, MTU 1500

unit-2-3:\*\*\*\*\*

MAC address 0015.c500.026f, MTU 1500

3 . unit-2-2는 프래그먼트된 패킷을 수신하고 리어셈블하며 흐름의 소유자가 됩니다.

<#root>

firepower#

cluster exec unit unit-2-2 show capture capccl packet-number 1 trace

11 packets captured

1: 20:13:58.231845 192.0.2.10 > 203.0.113.10 icmp: echo request

Phase: 1

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'inside'

Flow type: NO FLOW

I (2) received a FWD\_FRAG\_TO\_FRAG\_OWNER from (0).

Phase: 2

Type: CLUSTER-EVENT

Subtype:

Result: ALLOW

Config:

Additional Information:

Input interface: 'inside'

Flow type: NO FLOW

I (2) have reassembled a packet and am processing it.

Phase: 3

Type: CAPTURE

Subtype:

Result: ALLOW

Config:

Additional Information:

MAC Access list

Phase: 4

Type: ACCESS-LIST

Subtype:

Result: ALLOW

Config:

Implicit Rule

Additional Information:

MAC Access list

Phase: 5

Type: ROUTE-LOOKUP

Subtype: No ECMP load balancing

Result: ALLOW

Config:

Additional Information:

Destination is locally connected. No ECMP load balancing.

Found next-hop 203.0.113.10 using egress ifc outside(vrfid:0)

Phase: 6

Type: CLUSTER-EVENT

Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'inside'

Flow type: NO FLOW

I (2) am becoming owner

Phase: 7  
Type: ACCESS-LIST  
Subtype: log  
Result: ALLOW  
Config:  
access-group CSM\_FW\_ACL\_ global  
access-list CSM\_FW\_ACL\_ advanced trust ip any any rule-id 268435460 event-log flow-end  
access-list CSM\_FW\_ACL\_ remark rule-id 268435460: PREFILTER POLICY: igasimov\_prefilter1  
access-list CSM\_FW\_ACL\_ remark rule-id 268435460: RULE: r1  
Additional Information:

...

Phase: 19  
Type: FLOW-CREATION  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
New flow created with id 1719, packet dispatched to next module

...

Result:  
input-interface: cluster(vrfid:0)  
input-status: up  
input-line-status: up  
output-interface: outside(vrfid:0)  
output-status: up  
output-line-status: up

Action: allow

1 packet shown  
firepower#

cluster exec unit unit-2-2 show capture capccl packet-number 2 trace

11 packets captured

2: 20:13:58.231875  
Phase: 1

Type: CLUSTER-EVENT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Input interface: 'inside'

Flow type: NO FLOW

I (2) received a FWD\_FRAG\_TO\_FRAG\_OWNER from (0).

Result:  
input-interface: cluster(vrfid:0)  
input-status: up  
input-line-status: up  
Action: allow

1 packet shown

4. unit-2-2는 보안 정책에 따라 패킷을 허용하고 외부 인터페이스를 통해 사이트 2에서 사이트 1로 패킷을 전송합니다.

<#root>

firepower#

cluster exec unit unit-2-2 show cap capo

2 packets captured

1: 20:13:58.232058 802.1Q vlan#20 P0 192.0.2.10 > 203.0.113.10 icmp: echo request

2: 20:13:58.232058 802.1Q vlan#20 P0

관찰/주의 사항

- 디렉터 역할과 달리 프래그먼트 소유자는 특정 사이트 내에서 현지화할 수 없습니다. 프래그먼트 소유자는 원래 새 연결의 프래그먼트된 패킷을 수신하는 유닛에 의해 결정되며, 임의의 사이트에 위치할 수 있습니다.
- 프래그먼트 소유자도 연결 소유자가 될 수 있으므로 패킷을 목적지 호스트로 전달하려면 이그레스 인터페이스를 확인하고 목적지 호스트의 IP 및 MAC 주소 또는 다음 홉을 찾을 수 있어야 합니다. 이는 next-hop도 대상 호스트에 연결할 수 있어야 함을 전제로 합니다.

- 프래그먼트된 패킷을 리어셈블하기 위해 ASA/FTD는 명명된 각 인터페이스에 대해 IP 프래그먼트 리어셈블리 모듈을 유지합니다. IP 프래그먼트 리어셈블리 모듈의 운영 데이터를 표시하려면 show fragment 명령을 사용합니다.

```
<#root>
```

```
Interface: inside
Configuration:
```

```
Size: 200
```

```
, Chain: 24, Timeout: 5, Reassembly: virtual
Run-time stats: Queue: 0, Full assembly: 0
Drops: Size overflow: 0, Timeout: 0,
Chain overflow: 0, Fragment queue threshold exceeded: 0,
Small fragments: 0, Invalid IP len: 0,
Reassembly overlap: 0, Fraghead alloc failed: 0,
SGT mismatch: 0, Block alloc failed: 0,
Invalid IPV6 header: 0, Passenger flow assembly failed: 0
```

클러스터 구축에서 프래그먼트 소유자 또는 연결 소유자는 프래그먼트된 패킷을 프래그먼트 대기열에 넣습니다. 프래그먼트 큐 크기는 fragment size <size> <nameif> 명령으로 구성된 Size 카운터 값(기본값 200)에 의해 제한됩니다. 프래그먼트 대기열 크기가 Size의 2/3에 도달하면 프래그먼트 대기열 임계값이 초과된 것으로 간주되고 현재 프래그먼트 체인의 일부가 아닌 새 프래그먼트가 삭제됩니다. 이 경우 Fragment queue threshold exceeded가 증가하고 syslog 메시지 FTD-3-209006이 생성됩니다.

```
<#root>
```

```
firepower#
```

```
show fragment inside
```

```
Interface: inside
```

```
Configuration:
```

```
Size: 200
```

```
, Chain: 24, Timeout: 5, Reassembly: virtual
Run-time stats:
```

```
Queue: 133
```

```
, Full assembly: 0
```

```
Drops: Size overflow: 0, Timeout: 8178,
Chain overflow: 0,
```

```
Fragment queue threshold exceeded: 40802
```

```
,
Small fragments: 0, Invalid IP len: 0,
Reassembly overlap: 9673, Fraghead alloc failed: 0,
SGT mismatch: 0, Block alloc failed: 0,
Invalid IPV6 header: 0, Passenger flow assembly failed: 0
```

```
%FTD-3-209006: Fragment queue threshold exceeded, dropped TCP fragment from 192.0.2.10/21456 to 203.0.113.1
```

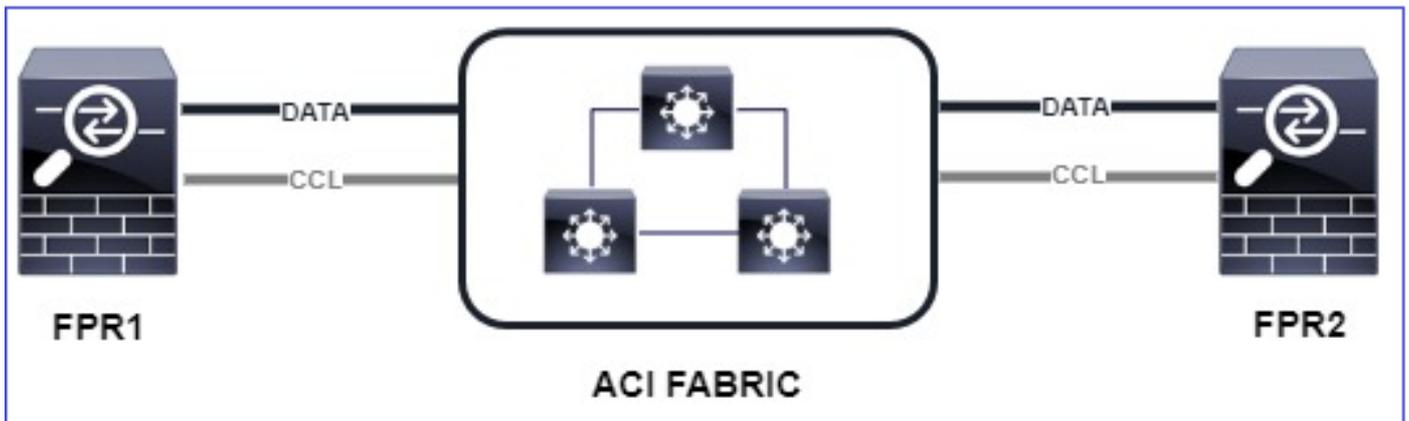
이를 해결하려면 Firepower Management Center > Devices > Device Management > [Edit Device] > Interfaces > [Interface] > Advanced > Security Configuration > Override Default Fragment Setting, save configuration and deploy policies에서 크기를 늘립니다. 그런 다음 show fragment 명령 출력 및 syslog 메시지 FTD-3-209006 발생에서 Queue 카운터를 모니터링합니다.

### ACI 문제

ACI 포드의 활성 L4 체크섬 확인으로 인해 클러스터를 통한 간헐적인 연결 문제

### 증상

- ACI Pod에 구축된 ASA/FTD 클러스터를 통한 간헐적인 연결 문제
- 클러스터에 유닛이 1개만 있는 경우 연결 문제가 관찰되지 않습니다.
- 한 클러스터 유닛에서 클러스터의 하나 이상의 다른 유닛으로 전송된 패킷은 FXOS 및 대상 유닛의 데이터 플레인 캡처에서 볼 수 없습니다.



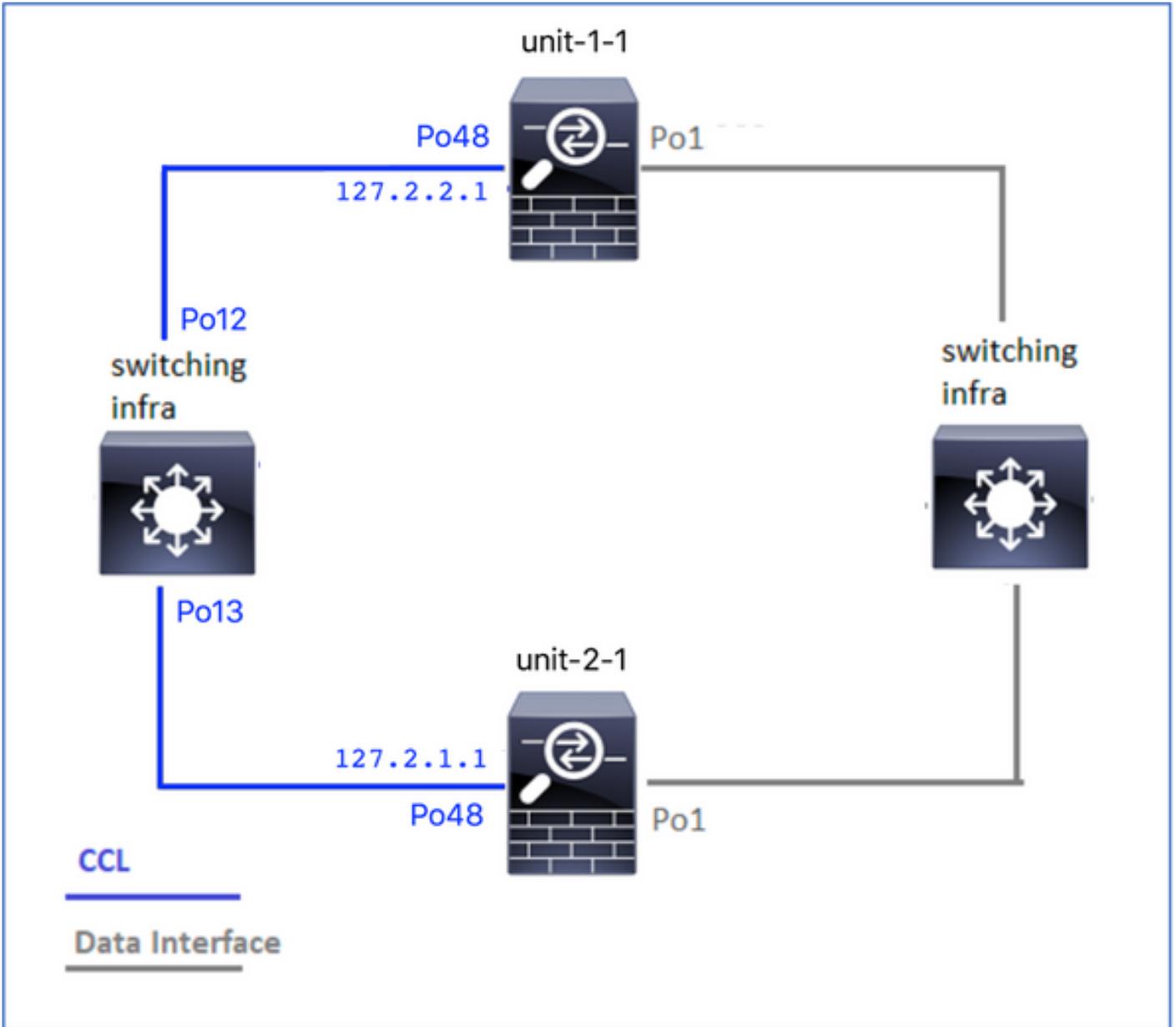
### 완화

- 클러스터 제어 링크를 통해 리디렉션된 트래픽에 올바른 L4 체크섬이 없으며 이는 정상적인 동작입니다. 클러스터 제어 링크 경로의 스위치는 L4 체크섬을 확인하지 않아야 합니다. L4 체크섬을 확인하는 스위치로 인해 트래픽이 삭제될 수 있습니다. ACI 패브릭 스위치 컨피그레이션을 확인하고 클러스터 제어 링크를 통해 수신된 패킷 또는 전송된 패킷에 대해 L4 체크섬이 수행되지 않는지 확인합니다.

### 클러스터 컨트롤 플레인 문제

유닛이 클러스터에 참가할 수 없음

CCL의 MTU 크기



## 증상

유닛이 클러스터에 가입할 수 없으며 이 메시지가 표시됩니다.

The SECONDARY has left the cluster because application configuration sync is timed out on this unit. Di...  
 Cluster disable is performing cleanup..done.  
 Unit unit-2-1 is quitting due to system failure for 1 time(s) (last failure is SECONDARY application co...  
 All data interfaces have been shutdown due to clustering being disabled. To recover either enable clust...

## 확인/완화

- FTD에서 show interface 명령을 사용하여 클러스터 제어 링크 인터페이스의 MTU가 데이터 인터페이스 MTU보다 100바이트 이상 높은지 확인합니다.

<#root>

```
firepower#
```

```
show interface
```

```
Interface
```

```
Port-channel1
```

```
"
```

```
Inside
```

```
", is up, line protocol is up  
Hardware is EtherSVI, BW 40000 Mbps, DLY 10 usec  
MAC address 3890.a5f1.aa5e,
```

```
MTU 9084
```

```
Interface
```

```
Port-channel48
```

```
"
```

```
cluster
```

```
", is up, line protocol is up  
Hardware is EtherSVI, BW 40000 Mbps, DLY 10 usec  
Description: Clustering Interface  
MAC address 0015.c500.028f,
```

```
MTU 9184
```

```
IP address 127.2.2.1, subnet mask 255.255.0.
```

- CCL에 구성된 MTU가 경로의 모든 디바이스에 올바르게 구성되었는지 확인하려면 size(크기) 옵션을 사용하여 CCL을 통해 ping을 수행합니다.

```
<#root>
```

```
firepower#
```

```
ping 127.2.1.1 size 9184
```

- 스위치에서 show interface 명령을 사용하여 MTU 컨피그레이션을 확인합니다

```
<#root>
```

```
Switch#
```

```
show interface
```

```
port-channel12
```

```
is up
admin state is up,
  Hardware: Port-Channel, address: 7069.5a3a.7976 (bia 7069.5a3a.7976)
```

MTU 9084

bytes, BW 40000000 Kbit , DLY 10 usec

port-channel13

```
is up
admin state is up,
  Hardware: Port-Channel, address: 7069.5a3a.7967 (bia 7069.5a3a.7967)
```

MTU 9084

bytes, BW 40000000 Kbit , DLY 10 use

## 클러스터 유닛 간의 인터페이스 불일치

### 증상

유닛이 클러스터에 가입할 수 없으며 이 메시지가 표시됩니다.

```
Interface mismatch between cluster primary and joining unit unit-2-1. unit-2-1 aborting cluster join.
Cluster disable is performing cleanup..done.
Unit unit-2-1 is quitting due to system failure for 1 time(s) (last failure is Internal clustering error)
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clustering
```

### 확인/완화

각 새시의 FCM GUI에 로그인하고 Interfaces(인터페이스) 탭으로 이동하여 모든 클러스터 멤버의 인터페이스 컨피그레이션이 동일한지 확인합니다.

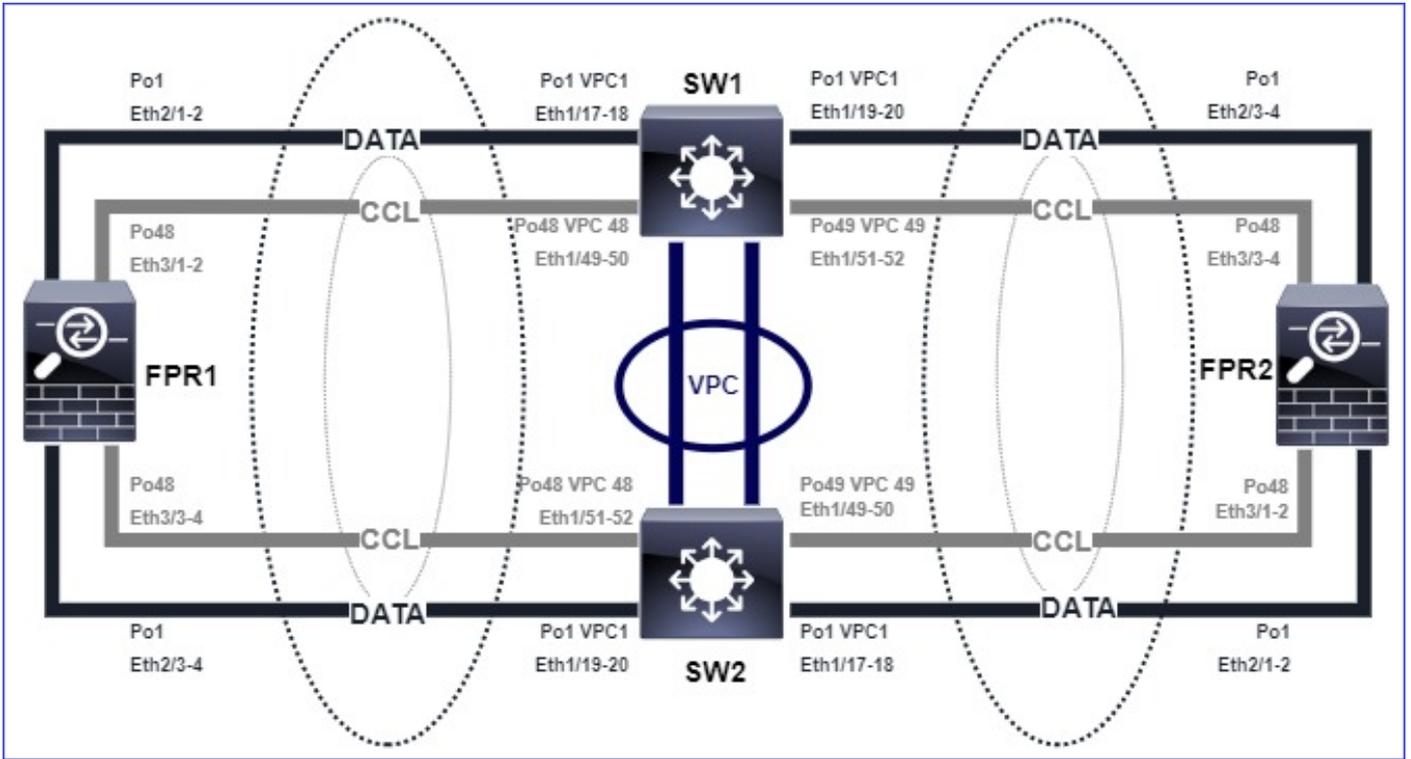
- 논리적 디바이스에 할당된 인터페이스
- 인터페이스의 관리 속도
- 인터페이스의 Admin 듀플렉스
- 인터페이스 상태

## 데이터/포트 채널 인터페이스 문제

### CCL에 대한 도달 가능성 문제로 인한 스플릿 브레인

### 증상

클러스터에는 여러 개의 제어 유닛이 있습니다. 다음 토폴로지를 고려하십시오.



새시 1:

<#root>

```
firepower# show cluster info
```

```
Cluster ftd_cluster1: On
Interface mode: spanned
```

```
This is "unit-1-1" in state PRIMARY
```

```
ID : 0
Site ID : 1
Version : 9.15(1)
Serial No.: FLM2103TU5H
CCL IP : 127.2.1.1
CCL MAC : 0015.c500.018f
Last join : 07:30:25 UTC Dec 14 2020
Last leave: N/A
Other members in the cluster:
Unit "unit-1-2" in state SECONDARY
ID : 1
Site ID : 1
Version : 9.15(1)
Serial No.: FLM2103TU4D
CCL IP : 127.2.1.2
CCL MAC : 0015.c500.019f
Last join : 07:30:26 UTC Dec 14 2020
Last leave: N/A
Unit "unit-1-3" in state SECONDARY
ID : 3
Site ID : 1
Version : 9.15(1)
```

Serial No.: FLM2102THJT  
CCL IP : 127.2.1.3  
CCL MAC : 0015.c500.016f  
Last join : 07:31:49 UTC Dec 14 2020  
Last leave: N/A

새시 2:

<#root>

firepower# show cluster info

Cluster ftd\_cluster1: On  
Interface mode: spanned

This is "unit-2-1" in state PRIMARY

ID : 4  
Site ID : 1  
Version : 9.15(1)  
Serial No.: FLM2103TUN1  
CCL IP : 127.2.2.1  
CCL MAC : 0015.c500.028f  
Last join : 11:21:56 UTC Dec 23 2020  
Last leave: 11:18:51 UTC Dec 23 2020  
Other members in the cluster:  
Unit "unit-2-2" in state SECONDARY  
ID : 2  
Site ID : 1  
Version : 9.15(1)  
Serial No.: FLM2102THR9  
CCL IP : 127.2.2.2  
CCL MAC : 0015.c500.029f  
Last join : 11:18:58 UTC Dec 23 2020  
Last leave: 22:28:01 UTC Dec 22 2020  
Unit "unit-2-3" in state SECONDARY  
ID : 5  
Site ID : 1  
Version : 9.15(1)  
Serial No.: FLM2103TUML  
CCL IP : 127.2.2.3  
CCL MAC : 0015.c500.026f  
Last join : 11:20:26 UTC Dec 23 2020  
Last leave: 22:28:00 UTC Dec 22 2020

확인

- ping 명령을 사용하여 제어 유닛의 CCL(Cluster Control Link) IP 주소 간의 연결을 확인할 수 있습니다.

<#root>

```
firepower# ping 127.2.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 127.2.1.1, timeout is 2 seconds:

?????

Success rate is 0 percent (0/5)

- ARP 테이블을 확인합니다.

<#root>

```
firepower# show arp
```

```
c1uster 127.2.2.3 0015.c500.026f 1
```

```
c1uster 127.2.2.2 0015.c500.029f 1
```

- 제어 유닛에서는 CCL 인터페이스의 캡처를 구성하고 확인합니다.

<#root>

```
firepower# capture capcc1 interface cluster
```

```
firepower# show capture capcc1 | i 127.2.1.1
```

```
2: 12:10:57.652310 arp who-has 127.2.1.1 tell 127.2.2.1
41: 12:11:02.652859 arp who-has 127.2.1.1 tell 127.2.2.1
74: 12:11:07.653439 arp who-has 127.2.1.1 tell 127.2.2.1
97: 12:11:12.654018 arp who-has 127.2.1.1 tell 127.2.2.1
126: 12:11:17.654568 arp who-has 127.2.1.1 tell 127.2.2.1
151: 12:11:22.655148 arp who-has 127.2.1.1 tell 127.2.2.1
174: 12:11:27.655697 arp who-has 127.2.1.1 tell 127.2.2.1
```

완화

- CCL 포트 채널 인터페이스가 스위치의 개별 포트 채널 인터페이스에 연결되었는지 확인합니다.
- Nexus 스위치에서 vPC(virtual port-channel)를 사용하는 경우 CCL 포트 채널 인터페이스가 서로 다른 vPC에 연결되어 있고 vPC 컨피그레이션에 장애 일관성 상태가 없는지 확인합니다.
- CCL 포트 채널 인터페이스가 동일한 브로드캐스트 도메인에 있고 CCL VLAN이 생성되어 인터페이스에서 허용되는지 확인합니다.

다음은 샘플 스위치 컨피그레이션입니다.

<#root>

Nexus#

show run int po48-49

```
interface port-channel48
description FPR1
```

```
switchport access vlan 48
```

```
vpc 48
```

```
interface port-channel49
description FPR2
```

```
switchport access vlan 48
```

```
vpc 49
```

Nexus#

show vlan id 48

VLAN Name Status Ports

-----

48 CCL active Po48, Po49, Po100, Eth1/53, Eth1/54

VLAN Type Vlan-mode

-----

48 enet CE

1 Po1 up success success 10,20

48 Po48 up success success 48

49 Po49 up success success 48

<#root>

Nexus1#

show vpc brief

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

vPC domain id : 1  
Peer status : peer adjacency formed ok  
vPC keep-alive status : peer is alive  
Configuration consistency status : success

Per-vlan consistency status : success

Type-2 consistency status : success

vPC role : primary  
Number of vPCs configured : 3  
Peer Gateway : Disabled  
Dual-active excluded VLANs : -  
Graceful Consistency Check : Enabled  
Auto-recovery status : Disabled  
Delay-restore status : Timer is off.(timeout = 30s)  
Delay-restore SVI status : Timer is off.(timeout = 10s)

vPC Peer-link status

-----  
id Port Status Active vlans  
-----

1 Po100 up 1,10,20,48-49,148

vPC status

-----  
id Port Status Consistency Reason Active vlans  
-----

1 Po1 up success success 10,20

48 Po48 up success success 48

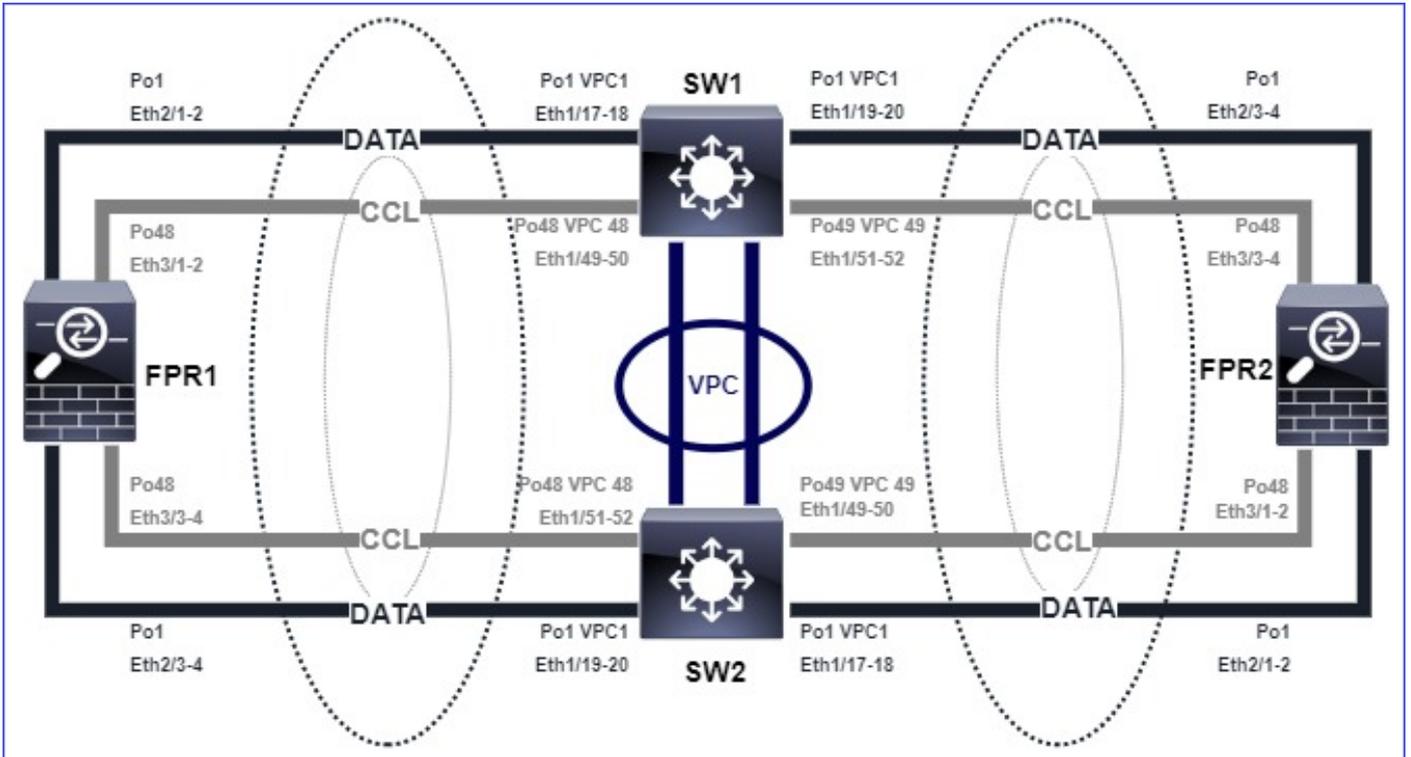
49 Po49 up success success 48

일시 중단된 데이터 포트 채널 인터페이스로 인해 클러스터가 비활성화되었습니다.

증상

하나 이상의 데이터 포트 채널 인터페이스가 일시 중단됩니다. 관리상 활성화된 데이터 인터페이스가 일시 중단되면 인터페이스 상태 확인 실패로 인해 동일한 새시의 모든 클러스터 유닛이 클러스터에서 제외됩니다.

다음 토폴로지를 고려하십시오.



## 확인

- 컨트롤 유닛 콘솔을 확인합니다.

<#root>

firepower#

Beginning configuration replication to

SECONDARY unit-2-2

End Configuration Replication to SECONDARY.

Asking SECONDARY unit

unit-2-2

to quit because it

failed interface health

check 4 times (last failure on

Port-channel1

). Clustering must be manually enabled on the unit to rejoin.

- 영향을 받는 유닛에서 show cluster history 및 show cluster info trace module hc 명령의 출력을 확인합니다.

<#root>

firepower# Unit is kicked out from cluster because of interface health check failure.

Cluster disable is performing cleanup..done.  
All data interfaces have been shutdown due to clustering being disabled. To recover either enable cluster  
Cluster unit unit-2-1 transitioned from SECONDARY to DISABLED

firepower#

show cluster history

```
=====
From State To State Reason
=====
```

12:59:37 UTC Dec 23 2020  
ONCALL SECONDARY\_COLD Received cluster control message

12:59:37 UTC Dec 23 2020  
SECONDARY\_COLD SECONDARY\_APP\_SYNC Client progression done

13:00:23 UTC Dec 23 2020  
SECONDARY\_APP\_SYNC SECONDARY\_CONFIG SECONDARY application configuration sync done

13:00:35 UTC Dec 23 2020  
SECONDARY\_CONFIG SECONDARY\_FILESYS Configuration replication finished

13:00:36 UTC Dec 23 2020  
SECONDARY\_FILESYS SECONDARY\_BULK\_SYNC Client progression done

13:01:35 UTC Dec 23 2020

SECONDARY\_BULK\_SYNC DISABLED Received control message DISABLE (interface health check failure)

<#root>

firepower#

show cluster info trace module hc

Dec 23 13:01:36.636 [INFO]cluster\_fsm\_clear\_np\_flows: The clustering re-enable timer is started to expire  
Dec 23 13:01:32.115 [INFO]cluster\_fsm\_disable: The clustering re-enable timer is stopped.

Dec 23 13:01:32.115 [INFO]Interface Port-channel1 is down

- fxos 명령 셸에서 show port-channel summary 명령의 출력을 확인합니다.

<#root>

FPR2(fxos)#

```
show port-channel summary
```

```
Flags: D - Down P - Up in port-channel (members)
I - Individual H - Hot-standby (LACP only)
s - Suspended r - Module-removed
S - Switched R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met
```

```
-----
Group Port-Channel Type Protocol Member Ports
-----
```

```
1 Po1(SD) Eth LACP Eth2/1(s) Eth2/2(s) Eth2/3(s) Eth2/4(s)
```

```
48 Po48(SU) Eth LACP Eth3/1(P) Eth3/2(P) Eth3/3(P) Eth3/4(P)
```

## 완화

- 모든 새시에 동일한 클러스터 그룹 이름과 비밀번호가 있는지 확인합니다.
- 포트 채널 인터페이스가 모든 새시와 스위치에서 동일한 듀플렉스/속도 컨피그레이션으로 관리적으로 활성화된 물리적 멤버 인터페이스를 가지고 있는지 확인합니다.
- Intra-site 클러스터에서는 모든 새시의 동일한 데이터 포트 채널 인터페이스가 스위치의 동일한 포트 채널 인터페이스에 연결되어야 합니다.
- Nexus 스위치에서 vPC(virtual port-channel)를 사용하는 경우 vPC 컨피그레이션에 실패한 일관성 상태가 없는지 확인합니다.
- 사이트 내 클러스터에서는 모든 새시의 동일한 데이터 포트 채널 인터페이스가 동일한 vPC에 연결되어 있는지 확인합니다.

## 클러스터 안정성 문제

### FXOS 역추적

### 증상

유닛이 클러스터를 나갑니다.

### 확인/완화

- show cluster history 명령을 사용하여 유닛이 언제 클러스터를 떠났는지 확인합니다

```
<#root>
```

```
firepower#
```

```
show cluster history
```

- 이 명령을 사용하여 FXOS에 역추적 기능이 있는지 확인합니다

<#root>

FPR4150#

connect local-mgmt

FPR4150 (local-mgmt)#

dir cores

- 유닛이 클러스터를 떠난 시점에 생성된 코어 파일을 수집하여 TAC에 제공합니다.

## 디스크 가득 참

클러스터 유닛의 /ngfw 파티션의 디스크 사용률이 94%에 도달하는 경우 유닛에서 클러스터를 종료합니다. 디스크 사용률 검사는 3초마다 수행됩니다.

<#root>

> show disk

```
Filesystem Size Used Avail Use% Mounted on
rootfs 81G 421M 80G 1% /
devtmpfs 81G 1.9G 79G 3% /dev
tmpfs 94G 1.8M 94G 1% /run
tmpfs 94G 2.2M 94G 1% /var/volatile
/dev/sda1 1.5G 156M 1.4G 11% /mnt/boot
/dev/sda2 978M 28M 900M 3% /opt/cisco/config
/dev/sda3 4.6G 88M 4.2G 3% /opt/cisco/platform/logs
/dev/sda5 50G 52M 47G 1% /var/data/cores
/dev/sda6 191G 191G 13M
```

100% /ngfw

cgroup\_root 94G 0 94G 0% /dev/cgroups

이 경우 show cluster history 출력에는 다음이 표시됩니다.

<#root>

15:36:10 UTC May 19 2021

PRIMARY Event: Primary unit unit-1-1 is quitting  
due to

diskstatus

Application health check failure, and  
primary's application state is down

또는

14:07:26 CEST May 18 2021

SECONDARY DISABLED Received control message DISABLE (application health check failure)

실패를 확인하는 또 다른 방법은 다음과 같습니다.

<#root>

firepower#

show cluster info health

Member ID to name mapping:

0 - unit-1-1(myself) 1 - unit-2-1

	0	1
Port-channel48	up	up
Ethernet1/1	up	up
Port-channel12	up	up
Port-channel13	up	up

Unit overall healthy healthy

Service health status:

	0	1
diskstatus (monitor on)	down	down
snort (monitor on)	up	up

Cluster overall healthy

또한 디스크가 ~100%인 경우, 일부 디스크 공간이 해제될 때까지 유닛이 클러스터에 다시 가입하는 데 어려움이 있을 수 있습니다.

오버플로 보호

각 클러스터 유닛에서는 5분마다 로컬 및 피어 유닛에서 CPU 및 메모리 사용률을 확인합니다. 사용률이 시스템 임계값보다 높은 경우(LINA CPU 50% 또는 LINA 메모리 59%) 다음 위치에 정보 메시지가 표시됩니다.

- Syslog(FTD-6-748008)
- log/cluster\_trace.log 파일을 참조하십시오. 예:

<#root>

firepower#

```
more log/cluster_trace.log | i CPU
```

```
May 20 16:18:06.614 [INFO][
```

```
CPU load 87%
```

```
| memory load 37%] of module 1 in chassis 1 (unit-1-1) exceeds overflow protection threshold [
```

```
CPU 50% | Memory 59%
```

```
]. System may be oversubscribed on member failure.
```

```
May 20 16:18:06.614 [INFO][CPU load 87% | memory load 37%] of chassis 1 exceeds overflow protection thr
```

```
May 20 16:23:06.644 [INFO][CPU load 84% | memory load 35%] of module 1 in chassis 1 (unit-1-1) exceeds o
```

이 메시지는 유닛 오류가 발생할 경우 다른 유닛 리소스가 초과 서브스크립션될 수 있음을 나타냅니다.

### 간소화된 모드

#### 6.3 이전 FMC 릴리스의 동작

- FMC에서 각 클러스터 노드를 개별적으로 등록합니다.
- 그런 다음 FMC에서 논리적 클러스터를 구성합니다.
- 새 클러스터 노드를 추가할 때마다 노드를 수동으로 등록해야 합니다.

#### 6.3 이후 FMC

- 간소화된 모드 기능을 사용하면 전체 클러스터를 FMC에 한 번에 등록할 수 있습니다(클러스터의 한 노드만 등록하면 됨).

최소 지원 관리자	관리되는 디바이스	최소 지원 관리되는 디바이스 버전 필요	참고
FMC 6.3	FP9300 및 FP4100에서만 FTD 클러스터	6.2.0	이는 FMC 기능에만 해당됩니다

 경고: FTD에서 클러스터가 형성되면 자동 등록이 시작될 때까지 기다려야 합니다. 클러스터 노드를 수동으로 등록(Add Device)하지 말고 Reconcile 옵션을 사용해야 합니다.

### 증상

#### 노드 등록 실패

- 어떤 이유로든 제어 노드 등록이 실패하면 클러스터는 FMC에서 삭제됩니다.

### 완화

어떤 이유로든 데이터 노드 등록이 실패하면 2가지 옵션이 있습니다.

1. 클러스터에 구축할 때마다 FMC는 등록해야 하는 클러스터 노드가 있는지 확인한 다음 이러한 노드에 대한 자동 등록을 시작합니다.
2. 클러스터 요약 탭(Devices(디바이스) > Device Management(디바이스 관리) > Cluster(클러스터) 탭 > View Cluster Status(클러스터 상태 보기) 링크)에서 Reconcile(조정) 옵션을 사용할 수 있습니다. Reconcile 작업이 트리거되면 FMC는 등록해야 하는 노드의 자동 등록을 시작합니다.

## 관련 정보

- [firepower 위협 방어를 위한 클러스터링](#)
- [firepower 4100/9300 새시용 ASA 클러스터](#)
- [firepower 4100/9300 새시의 클러스터링 정보](#)
- [Firepower NGFW 클러스터링 심층 분석 - BRKSEC-3032](#)
- [Firepower 방화벽 캡처를 분석하여 네트워크 문제를 효과적으로 해결](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.