

# Firepower 방화벽 캡처를 분석하여 네트워크 문제를 효과적으로 해결

## 목차

---

### [소개](#)

#### [사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

#### [배경 정보](#)

#### [NGFW 제품군에 대한 캡처를 수집하고 내보내는 방법](#)

[FXOS 캡처 수집](#)

[FTD Lina 캡처 활성화 및 수집](#)

[FTD Snort 캡처 활성화 및 수집](#)

#### [문제 해결](#)

##### [사례 1. 이그레스 인터페이스에 TCP SYN 없음](#)

[캡처 분석](#)

[권장 작업](#)

[가능한 원인 및 권장 조치 요약](#)

##### [사례 2. 클라이언트의 TCP SYN, 서버의 TCP RST](#)

[캡처 분석](#)

[권장 작업](#)

##### [사례 3. TCP 3-Way 핸드셰이크 + 한 엔드포인트의 RST](#)

[캡처 분석](#)

[3.1 - TCP 3-way 핸드셰이크 + 클라이언트에서 지연된 RST](#)

[권장 작업](#)

[3.2 - TCP 3-way 핸드셰이크 + 클라이언트에서 지연된 FIN/ACK + 서버에서 지연된 RST](#)

[권장 작업](#)

[3.3 - TCP 3-way 핸드셰이크 + 클라이언트에서 지연된 RST](#)

[권장 작업](#)

[3.4 - TCP 3-way 핸드셰이크 + 서버에서 즉시 RST](#)

[권장 작업](#)

##### [사례 4. 클라이언트의 TCP RST](#)

[캡처 분석](#)

[권장 작업](#)

##### [사례 5. 느린 TCP 전송\(시나리오 1\)](#)

[시나리오 1. 저속 전송](#)

[캡처 분석](#)

[권장 작업](#)

[시나리오 2. 빠른 전송](#)

##### [사례 6. 느린 TCP 전송\(시나리오 2\)](#)

[캡처 분석](#)

[권장 작업](#)

##### [사례 7. TCP 연결 문제\(패킷 손상\)](#)

[캡처 분석](#)

[권장 작업](#)

[사례 8. UDP 연결 문제\(누락된 패킷\)](#)

[캡처 분석](#)

[권장 작업](#)

[사례 9. HTTPS 연결 문제\(시나리오 1\)](#)

[캡처 분석](#)

[권장 작업](#)

[사례 10. HTTPS 연결 문제\(시나리오 2\)](#)

[캡처 분석](#)

[권장 작업](#)

[사례 11. IPv6 연결 문제](#)

[캡처 분석](#)

[권장 작업](#)

[사례 12. 간헐적 연결 문제\(ARP 중독\)](#)

[캡처 분석](#)

[권장 작업](#)

[사례 13. CPU 호그를 유발하는 SNMP OID\(Object Identifier\) 식별](#)

[캡처 분석](#)

[권장 작업](#)

[관련 정보](#)

---

## 소개

이 문서에서는 네트워크 문제를 효과적으로 해결하기 위한 다양한 패킷 캡처 분석 기술에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Firepower 플랫폼 아키텍처
- NGFW 로그
- NGFW 패킷 추적기

또한 패킷 캡처를 분석하기 전에 다음 요구 사항을 충족하는 것이 좋습니다.

- 프로토콜 작업 파악 - 캡처된 프로토콜의 작동 방식을 모르는 경우 패킷 캡처 확인을 시작하지 마십시오.
- 토폴로지 파악 - 전송 디바이스를 엔드 투 엔드로 알아야 합니다. 이 방법이 불가능한 경우 업 스트림 및 다운스트림 디바이스라도 알아야 합니다.
- 어플라이언스 파악 - 디바이스에서 패킷을 처리하는 방법, 관련된 인터페이스(인그레스/이그레스)는 무엇인지, 디바이스 아키텍처는 무엇인지, 다양한 캡처 포인트는 무엇인지 알아야 합니다.
- 컨피그레이션 파악 - 디바이스에서 패킷 플로우를 처리하는 방법을 알아야 합니다.
  - 라우팅/이그레스 인터페이스
  - 정책 적용됨

- NAT(Network Address Translation)
- 사용 가능한 툴 파악 - 캡처와 함께 다른 툴 및 기술(로깅 및 추적기 등)을 적용하고 필요한 경우 캡처된 패킷과 상호 연결할 준비가 된 것이 좋습니다

## 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- 대부분의 시나리오는 FTD 소프트웨어 6.5.x를 실행하는 FP4140을 기반으로 합니다.
- 소프트웨어 6.5.x를 실행하는 FMC

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

패킷 캡처는 현재 제공되는 가장 간과된 문제 해결 툴 중 하나입니다. 매일 Cisco TAC는 캡처된 데이터 분석으로 많은 문제를 해결합니다.

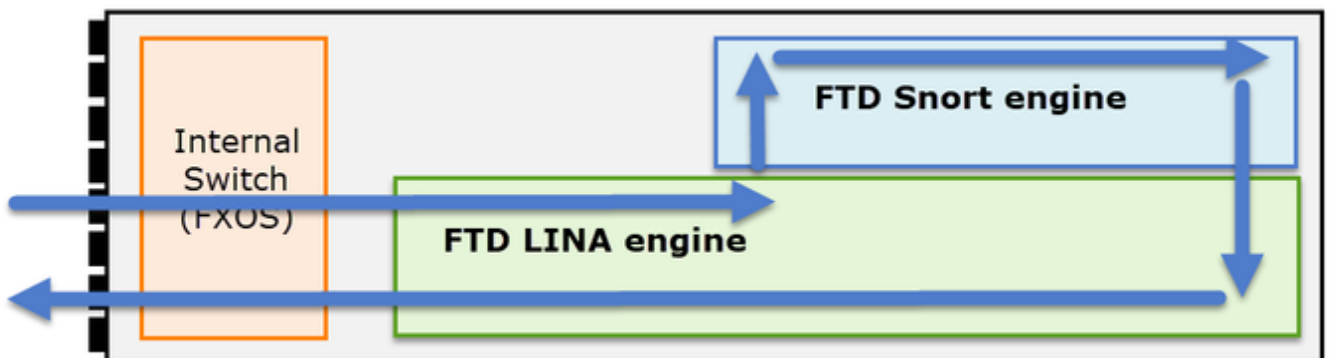
이 문서의 목표는 주로 패킷 캡처 분석을 기반으로 네트워크 및 보안 엔지니어가 일반적인 네트워크 문제를 식별하고 해결할 수 있도록 지원하는 것입니다.

이 문서에 제시된 모든 시나리오는 Cisco TAC(Technical Assistance Center)에서 볼 수 있는 실제 사용자 사례를 기반으로 합니다.

이 문서에서는 Cisco NGFW(Next-Generation Firewall) 관점에서 패킷 캡처를 다루지만, 다른 디바이스 유형에도 동일한 개념이 적용됩니다.

## NGFW 제품군에 대한 캡처를 수집하고 내보내는 방법

firepower 어플라이언스(1xxx, 21xx, 41xx, 93xx) 및 FTD(Firepower Threat Defense) 애플리케이션의 경우 그림과 같이 패킷 처리를 시각화할 수 있습니다.



1. 패킷은 인그레스 인터페이스로 들어가고 샤페 내부 스위치에 의해 처리됩니다.
2. 패킷은 주로 L3/L4 검사를 수행하는 FTD Lina 엔진에 들어갑니다.

3. 정책에 따라 패킷이 Snort 엔진에 의해 검사되어야 하는 경우(주로 L7 검사).
4. Snort 엔진이 패킷에 대한 판정을 반환합니다.
5. LINA 엔진은 Snort 판정을 기반으로 패킷을 삭제하거나 전달합니다.
6. 패킷이 내부 새시 스위치를 통해 새시를 이그레스(egress)합니다.

표시된 아키텍처를 기반으로 FTD 캡처는 세 가지 서로 다른 위치에서 수행할 수 있습니다.

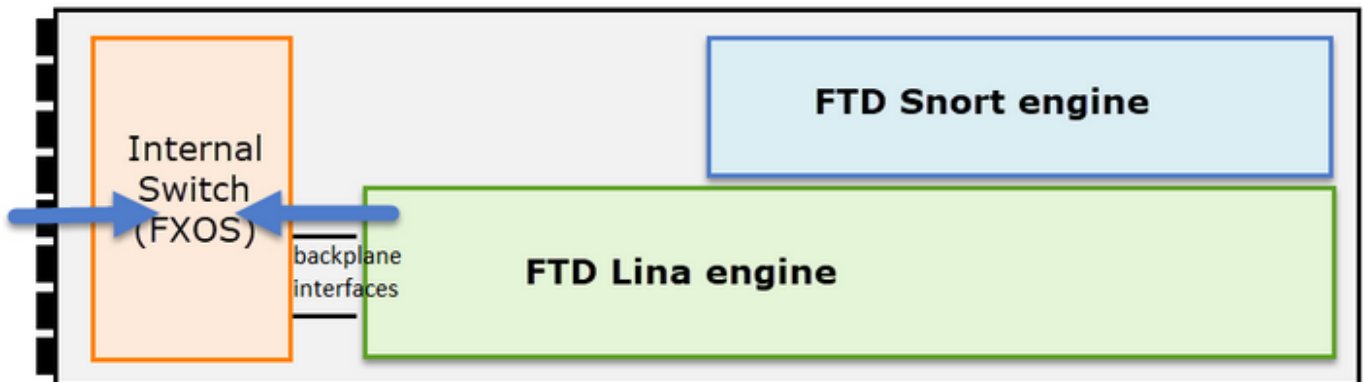
- FXOS
- FTD Lina 엔진
- FTD Snort 엔진

### FXOS 캡처 수집

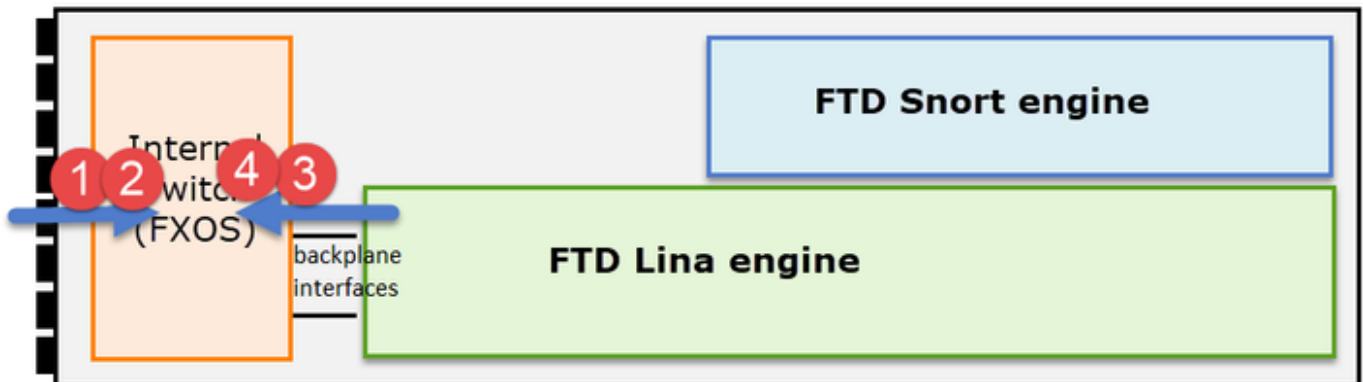
이 문서에서는 이 프로세스에 대해 설명합니다.

[https://www.cisco.com/c/en/us/td/docs/security/firepower/fxos/fxos271/web-guide/b\\_GUI\\_FXOS\\_ConfigGuide\\_271/troubleshooting.html#concept\\_E8823CC63C934A909BBC0DF12F](https://www.cisco.com/c/en/us/td/docs/security/firepower/fxos/fxos271/web-guide/b_GUI_FXOS_ConfigGuide_271/troubleshooting.html#concept_E8823CC63C934A909BBC0DF12F)

FXOS 캡처는 여기 이미지에 표시된 내부 스위치 관점에서 인그레스 방향으로만 수행할 수 있습니다.




이 슬라이드에는 내부 스위치 아키텍처로 인해 방향당 2개의 캡처 지점이 나와 있습니다.



포인트 2, 3, 4의 캡처된 패킷에는 VNTag(가상 네트워크 태그)가 있습니다.

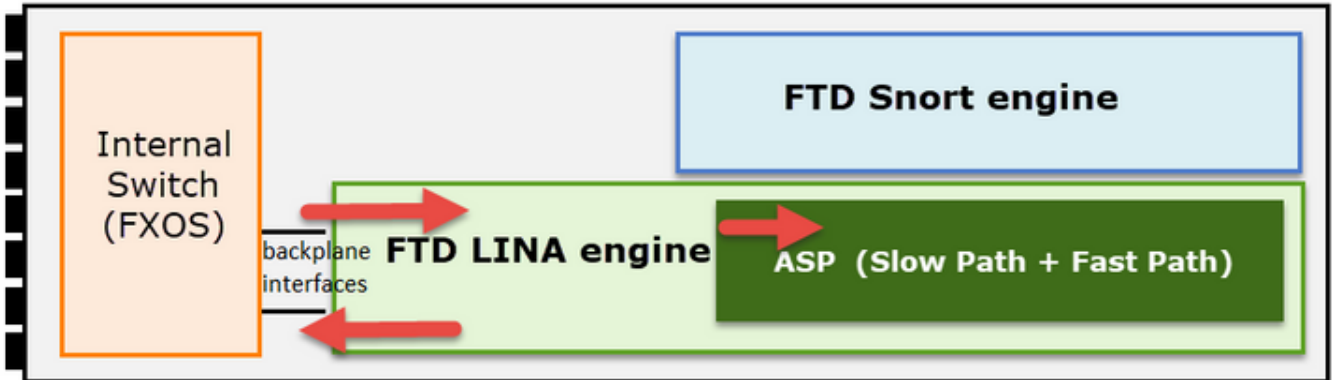
참고: FXOS 새시 레벨 캡처는 FP41xx 및 FP93xx 플랫폼에서만 사용할 수 있습니다. FP1xxx

 및 FP21xx는 이 기능을 제공하지 않습니다.

## FTD Lina 캡처 활성화 및 수집

주요 캡처 지점:

- 인그레스 인터페이스
- 이그레스 인터페이스
- ASP(Accelerated Security Path)



FMC UI(Firepower Management Center User Interface) 또는 FTD CLI를 사용하여 FTD Lina 캡처를 활성화하고 수집할 수 있습니다.

INSIDE 인터페이스의 CLI에서 캡처를 활성화합니다.

```
<#root>
```

```
firepower#
```

```
capture CAPI interface INSIDE match icmp host 192.168.103.1 host 192.168.101.1
```

이 캡처는 양방향의 IP 192.168.103.1과 192.168.101.1 간의 트래픽과 일치합니다.

FTD Lina 엔진에서 삭제된 모든 패킷을 보려면 ASP 캡처를 활성화합니다.

```
<#root>
```

```
firepower#
```

```
capture ASP type asp-drop all
```

FTP 서버로 FTD Lina 캡처 내보내기:

```
<#root>
```

```
firepower#
```

```
copy /pcap capture:CAPI ftp://ftp_username:ftp_password@192.168.78.73/CAPI.pcap
```

FTD Lina 캡처를 TFTP 서버로 내보냅니다.

```
<#root>
```

```
firepower#
```

```
copy /pcap capture:CAPI tftp://192.168.78.73
```

FMC 6.2.x 버전에서처럼 FMC UI에서 FTD Lina 캡처를 활성화하고 수집할 수 있습니다.

FMC 관리 방화벽에서 FTD 캡처를 수집하는 또 다른 방법은 다음과 같습니다.

1단계

LINA 또는 ASP 캡처의 경우 캡처를 FTD 디스크에 복사합니다.

```
<#root>
```

```
firepower#
```

```
copy /pcap capture:capin disk0:capin.pcap
```

```
Source capture name [capin]?
```

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

```
!!!!
```

2단계

expert 모드로 이동하여 저장된 캡처를 찾는 다음 /ngfw/var/common 위치에 복사합니다.

```
<#root>
```

```
firepower#
```

```
Console connection detached.
```

```
>
```

```
expert
```

```
admin@firepower:~$
```

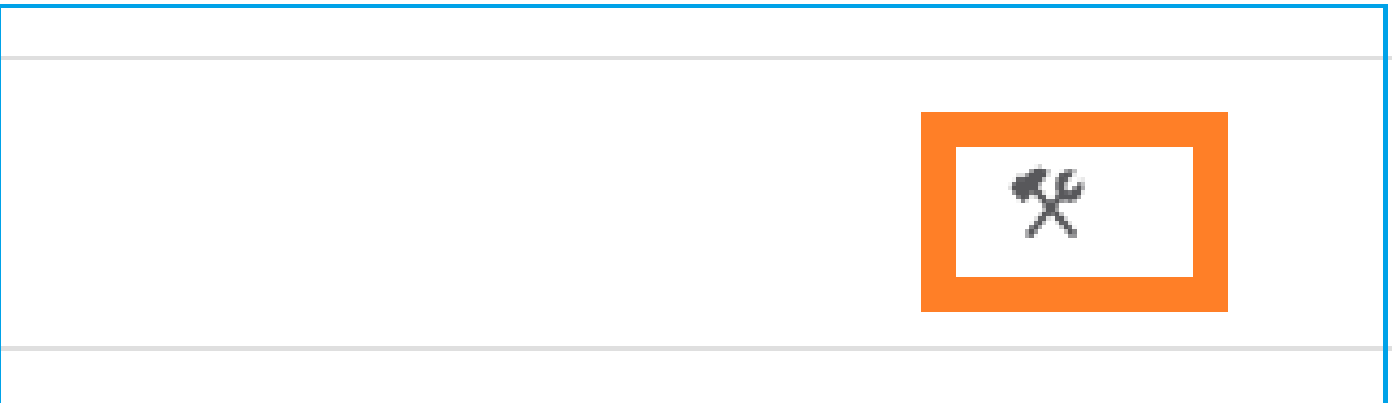
```
sudo su
```

```
Password:
```

```
root@firepower:/home/admin#  
cd /mnt/disk0  
root@firepower:/mnt/disk0#  
ls -al | grep pcap  
-rwxr-xr-x 1 root root    24 Apr 26 18:19 CAPI.pcap  
-rwxr-xr-x 1 root root 30110 Apr  8 14:10  
capin.pcap  
-rwxr-xr-x 1 root root  6123 Apr  8 14:11 capin2.pcap  
root@firepower:/mnt/disk0#  
cp capin.pcap /ngfw/var/common
```

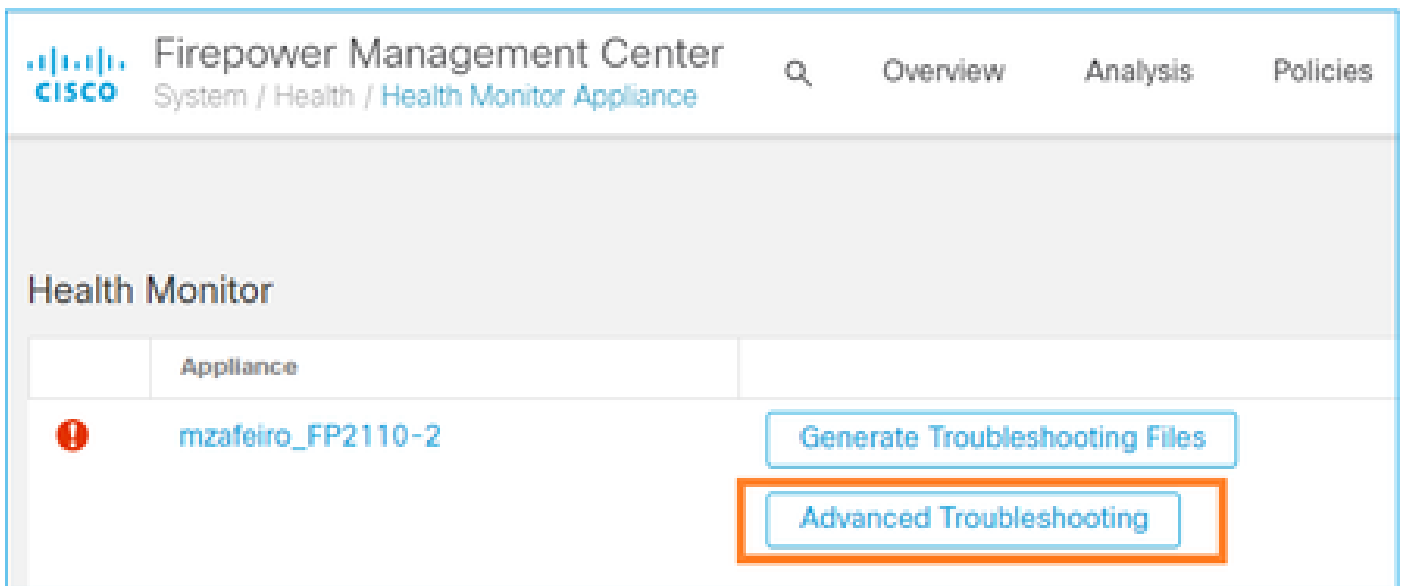
### 3단계

FTD를 관리하는 FMC에 로그인하고 Devices(디바이스) > Device Management(디바이스 관리)로 이동합니다. FTD 디바이스를 찾고 Troubleshoot(문제 해결) 아이콘을 선택합니다.

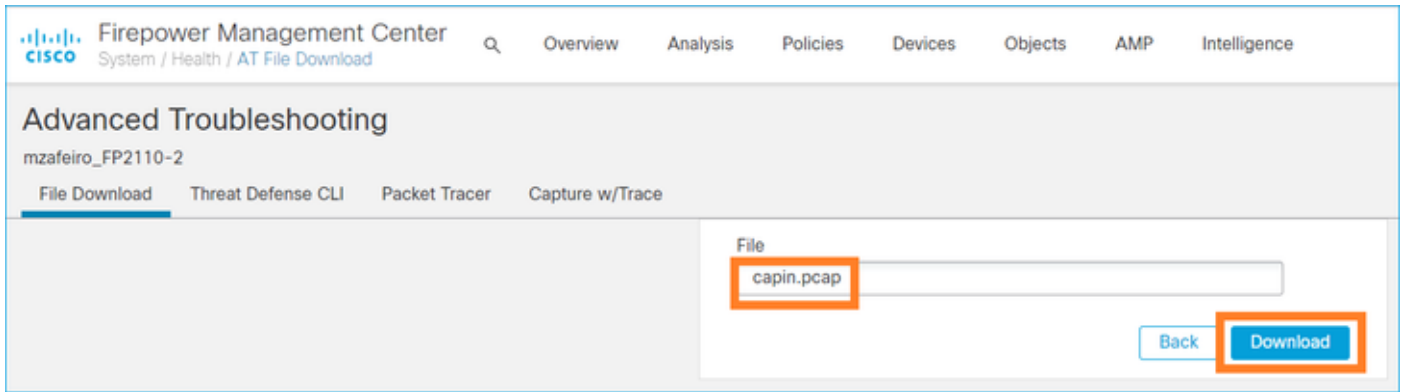


### 4단계

Advanced Troubleshooting(고급 트러블슈팅)을 선택합니다.



캡처 파일 이름을 지정하고 다운로드:

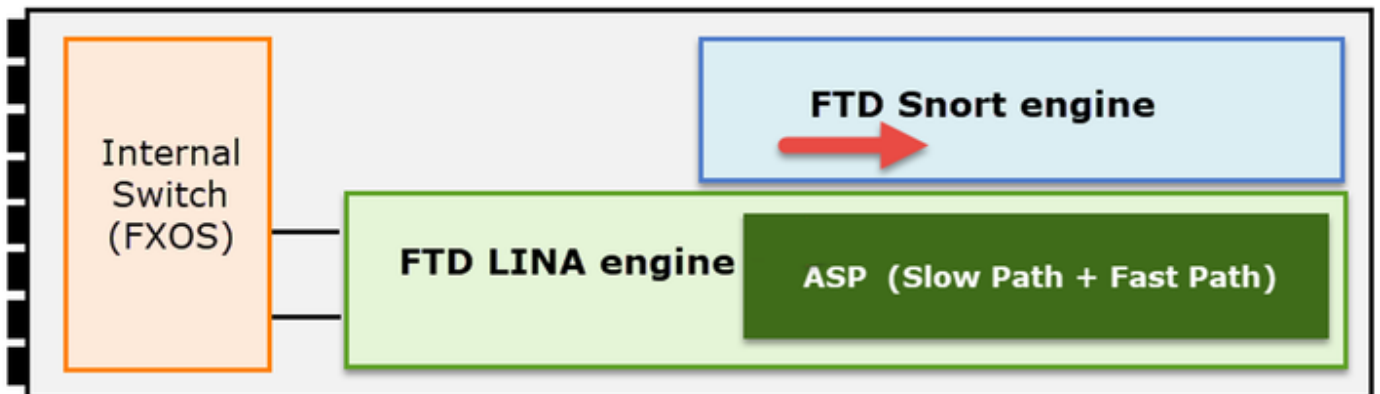


FMC UI에서 캡처를 활성화/수집하는 방법에 대한 자세한 예는 이 문서를 참조하십시오.

<https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/212474-working-with-firepower-threat-defense-f.html>

## FTD Snort 캡처 활성화 및 수집

여기 이미지에 캡처 지점이 표시됩니다.



Snort 레벨 캡처 활성화:

```
<#root>
```

```
>
```

```
capture-traffic
```

```
Please choose domain to capture traffic from:
```

```
0 - br1
```

```
1 - Router
```

```
Selection?
```

```
1
```

```
Please specify tcpdump options desired.
```

```
(or enter '?' for a list of supported options)
```

```
Options:
```



```
-n host 192.168.101.1
```

capture.pcap 이름을 가진 파일에 캡처를 쓰고 FTP를 통해 원격 서버에 복사하려면 다음을 수행합니다.

```
<#root>
```

```
>
```

```
capture-traffic
```

```
Please choose domain to capture traffic from:
```

```
0 - br1
```

```
1 - Router
```

```
Selection?
```

```
1
```

```
Please specify tcpdump options desired.
```

```
(or enter '?' for a list of supported options)
```

```
Options:
```

```
-w capture.pcap host 192.168.101.1
```

```
CTRL + C <- to stop the capture
```

```
>
```

```
file copy 10.229.22.136 ftp / capture.pcap
```

```
Enter password for ftp@10.229.22.136:
```

```
Copying capture.pcap
```

```
Copy successful.
```

```
>
```

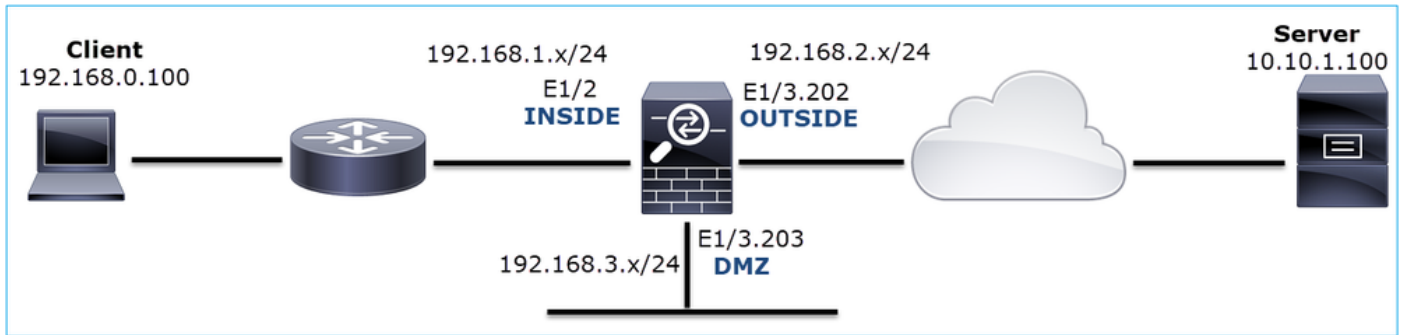
다른 캡처 필터를 포함하는 Snort 레벨 캡처 예제를 보려면 다음 문서를 확인하십시오.

<https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/212474-working-with-firepower-threat-defense-f.html>

## 문제 해결

# 사례 1. 이그레스 인터페이스에 TCP SYN 없음

토폴로지는 다음 이미지에 표시됩니다.



문제 설명: HTTP가 작동하지 않음

영향을 받는 흐름:

소스 IP: 192.168.0.100

Dst IP: 10.10.1.100

프로토콜: TCP 80

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

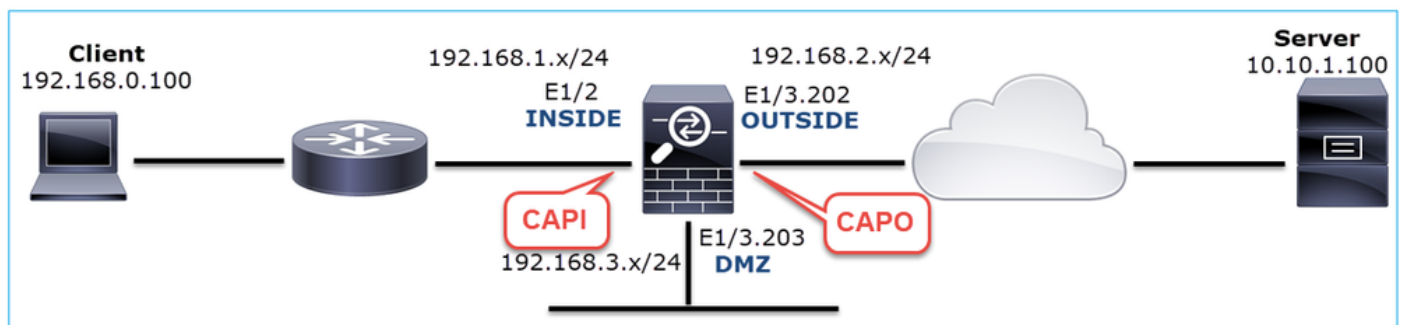
<#root>

firepower#

```
capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100
```

firepower#

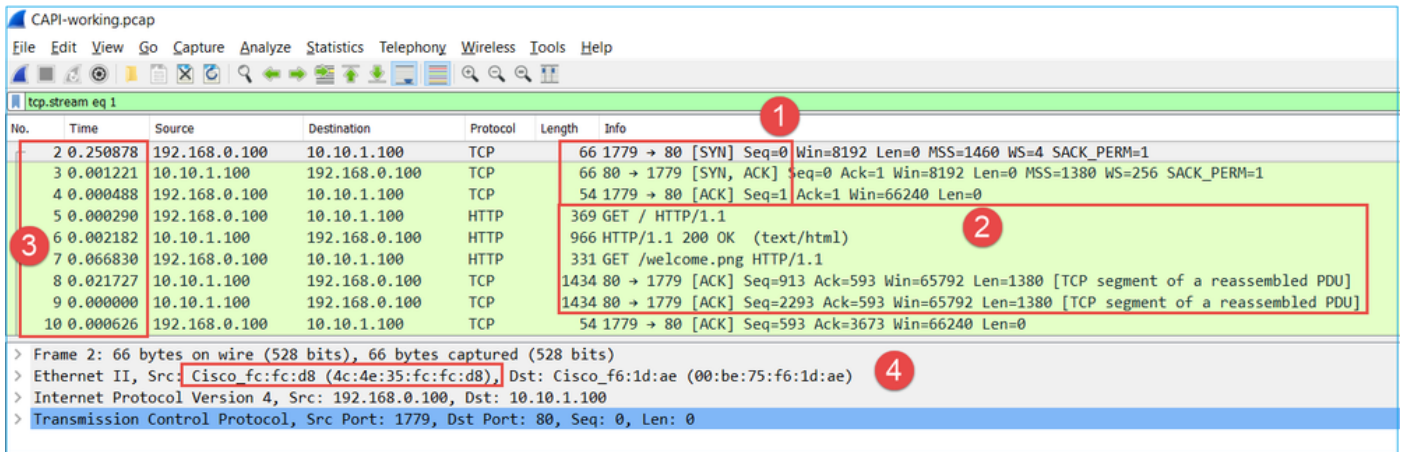
```
capture CAPO int OUTSIDE match ip host 192.168.0.100 host 10.10.1.100
```



캡처 - 기능 시나리오:

기존 요소로서 기능 시나리오의 캡처를 사용하는 것은 항상 매우 유용합니다.

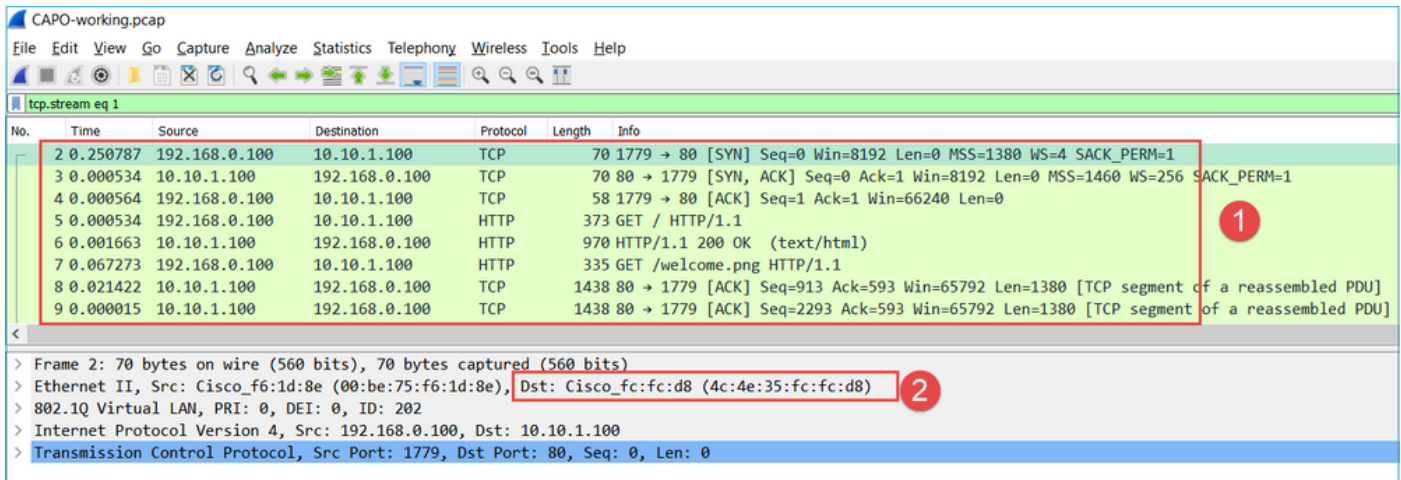
NGFW INSIDE 인터페이스에서 캡처한 내용은 다음과 같습니다.



요점:

1. TCP 3-way 핸드셰이크.
2. 양방향 데이터 교환.
3. 패킷 간의 시간 차이를 기준으로 패킷 간의 지연 없음
4. 소스 MAC는 올바른 다운스트림 디바이스입니다.

NGFW OUTSIDE 인터페이스에서 캡처한 내용은 다음 이미지에 나와 있습니다.



요점:

1. CAPI 캡처와 동일한 데이터.
2. 대상 MAC는 올바른 업스트림 디바이스입니다.

캡처 - 작동하지 않는 시나리오

디바이스 CLI에서 캡처는 다음과 같습니다.

```

<#root>
firepower#
show capture
    
```

```
capture CAPI type raw-data interface INSIDE
```

```
[Capturing - 484 bytes]
```

```
match ip host 192.168.0.100 host 10.10.1.100  
capture CAPO type raw-data interface OUTSIDE
```

```
[Capturing - 0 bytes]
```

```
match ip host 192.168.0.100 host 10.10.1.100
```

CAPI 내용:

```
<#root>
```

```
firepower#
```

```
show capture CAPI
```

```
6 packets captured
```

```
1: 11:47:46.911482 192.168.0.100.3171 > 10.10.1.100.80:
```

```
s
```

```
1089825363:1089825363(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
```

```
2: 11:47:47.161902 192.168.0.100.3172 > 10.10.1.100.80:
```

```
s
```

```
3981048763:3981048763(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
```

```
3: 11:47:49.907683 192.168.0.100.3171 > 10.10.1.100.80:
```

```
s
```

```
1089825363:1089825363(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
```

```
4: 11:47:50.162757 192.168.0.100.3172 > 10.10.1.100.80:
```

```
s
```

```
3981048763:3981048763(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
```

```
5: 11:47:55.914640 192.168.0.100.3171 > 10.10.1.100.80:
```

```
s
```

```
1089825363:1089825363(0) win 8192 <mss 1460,nop,nop,sackOK>
```

```
6: 11:47:56.164710 192.168.0.100.3172 > 10.10.1.100.80:
```

```
s
```

```
3981048763:3981048763(0) win 8192 <mss 1460,nop,nop,sackOK>
```

```
<#root>
```

```
firepower#
```

```
show capture CAPO
```

0 packet captured

0 packet shown

다음은 Wireshark의 CAPI 캡처 이미지입니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.100	10.10.1.100	TCP	66	3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.250420	192.168.0.100	10.10.1.100	TCP	66	3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	2.745781	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4	0.255074	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
5	5.751883	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
6	0.250070	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
 > Ethernet II, Src: Cisco\_fc:fc:d8 (4c:4e:35:fc:fc:d8), Dst: Cisco\_f6:1d:ae (00:be:75:f6:1d:ae)  
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100  
 > Transmission Control Protocol, Src Port: 3171, Dst Port: 80, Seq: 0, Len: 0

요점:

1. TCP SYN 패킷만 표시됩니다(TCP 3-way 핸드셰이크 없음).
2. 설정할 수 없는 2개의 TCP 세션(소스 포트 3171 및 3172)이 있습니다. 소스 클라이언트가 TCP SYN 패킷을 다시 전송합니다. 이러한 재전송된 패킷은 Wireshark에 의해 TCP 재전송으로 식별됩니다.
3. TCP 재전송은 3초 ~ 6초 간격으로 발생합니다.
4. 소스 MAC 주소는 올바른 다운스트림 디바이스에서 가져옵니다.

2개의 캡처를 바탕으로 다음과 같은 결론을 내릴 수 있습니다.

- 특정 5튜플 패킷(src/dst IP, src/dst port, protocol)이 예상 인터페이스(INSIDE)의 방화벽에 도착합니다.
- 패킷이 필요한 인터페이스(OUTSIDE)의 방화벽을 벗어나지 않습니다.

권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. 에뮬레이트된 패킷의 추적을 확인합니다.

패킷 추적기 툴을 사용하여 패킷이 방화벽에 의해 처리되는 방식을 확인합니다. 패킷이 방화벽 액세스 정책에 의해 삭제되는 경우 에뮬레이트된 패킷의 추적은 이 출력과 유사합니다.

<#root>

firepower#

packet-tracer input INSIDE tcp 192.168.0.100 11111 10.10.1.100 80

Phase: 1  
 Type: CAPTURE  
 Subtype:  
 Result: ALLOW

Config:  
Additional Information:  
MAC Access list

Phase: 2  
Type: ACCESS-LIST  
Subtype:  
Result: ALLOW  
Config:  
Implicit Rule  
Additional Information:  
MAC Access list

Phase: 3  
Type: ROUTE-LOOKUP  
Subtype: Resolve Egress Interface  
Result: ALLOW  
Config:  
Additional Information:  
found next-hop 192.168.2.72 using egress ifc OUTSIDE

Phase: 4  
Type: ACCESS-LIST  
Subtype: log  
Result: DROP

Config:  
access-group CSM\_FW\_ACL\_ global  
access-list CSM\_FW\_ACL\_ advanced deny ip any any rule-id 268439946 event-log flow-start  
access-list CSM\_FW\_ACL\_ remark rule-id 268439946: ACCESS POLICY: FTD\_Policy - Default  
access-list CSM\_FW\_ACL\_ remark rule-id 268439946: L4 RULE: DEFAULT ACTION RULE  
Additional Information:

Result:  
input-interface: INSIDE  
input-status: up  
input-line-status: up  
output-interface: OUTSIDE  
output-status: up  
output-line-status: up  
Action: drop

Drop-reason: (acl-drop) Flow is denied by configured rule, Drop-location: frame 0x00005647a4f4b120 flow

작업 2. 라이브 패킷의 추적을 확인합니다.

실제 TCP SYN 패킷이 방화벽에 의해 처리되는 방식을 확인하려면 패킷 추적을 활성화합니다. 기본적으로 처음 50개의 인그레스 패킷만 추적됩니다.

<#root>

firepower#

```
capture CAPI trace
```

캡처 버퍼를 지웁니다.

```
<#root>
```

```
firepower#
```

```
clear capture /all
```

패킷이 방화벽 액세스 정책에 의해 삭제된 경우 추적은 이 출력과 비슷합니다.

```
<#root>
```

```
firepower#
```

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

```
6 packets captured
```

```
1: 12:45:36.279740 192.168.0.100.3630 > 10.10.1.100.80: S 2322685377:2322685377(0) win 8192 <m
```

```
Phase: 1
```

```
Type: CAPTURE
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
MAC Access list
```

```
Phase: 2
```

```
Type: ACCESS-LIST
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Implicit Rule
```

```
Additional Information:
```

```
MAC Access list
```

```
Phase: 3
```

```
Type: ROUTE-LOOKUP
```

```
Subtype: Resolve Egress Interface
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
found next-hop 192.168.2.72 using egress ifc OUTSIDE
```

```
Phase: 4
```

```
Type: ACCESS-LIST
```

```
Subtype: log
```

```
Result: DROP
```

```
Config:
```

```
access-group CSM_FW_ACL_ global
access-list CSM_FW_ACL_ advanced deny ip any any rule-id 268439946 event-log flow-start
access-list CSM_FW_ACL_ remark rule-id 268439946: ACCESS POLICY: FTD_Policy - Default
access-list CSM_FW_ACL_ remark rule-id 268439946: L4 RULE: DEFAULT ACTION RULE
Additional Information:
```

```
Result:
input-interface: INSIDE
input-status: up
input-line-status: up
output-interface: OUTSIDE
output-status: up
output-line-status: up
Action: drop
```

```
Drop-reason: (acl-drop) Flow is denied by configured rule, Drop-location: frame 0x00005647a4f4b120 flow
```

```
1 packet shown
```

작업 3. FTD Lina 로그를 확인합니다.

FMC를 통해 FTD에서 Syslog를 구성하려면 다음 문서를 확인하십시오.

<https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/200479-Configure-Logging-on-FTD-via-FMC.html>

FTD Lina 로그에 대해 외부 Syslog 서버를 구성하는 것이 좋습니다. 구성된 원격 Syslog 서버가 없는 경우, 문제를 해결하는 동안 방화벽에서 로컬 버퍼 로그를 활성화합니다. 이 예에 표시된 로그 컨피그레이션은 좋은 시작점입니다.

```
<#root>
```

```
firepower#
```

```
show run logging
```

```
...
logging enable
logging timestamp
logging buffer-size 1000000
logging buffered informational
```

터미널 페이지를 제어하려면 터미널 페이지를 24행으로 설정합니다.

```
<#root>
```

```
firepower#
```



캡처 버퍼를 지웁니다.

```
<#root>
```

```
firepower#
```

```
clear logging buffer
```

연결을 테스트하고 파서 필터로 로그를 확인합니다. 이 예에서는 패킷이 방화벽 액세스 정책에 의해 삭제됩니다.

```
<#root>
```

```
firepower#
```

```
show logging | include 10.10.1.100
```

```
Oct 09 2019 12:55:51: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3696 dst OUTSIDE:10.10.1.100/80
Oct 09 2019 12:55:51: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3697 dst OUTSIDE:10.10.1.100/80
Oct 09 2019 12:55:54: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3696 dst OUTSIDE:10.10.1.100/80
Oct 09 2019 12:55:54: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3697 dst OUTSIDE:10.10.1.100/80
```

작업 4. ASP가 삭제하는 방화벽을 확인합니다.

패킷이 방화벽에 의해 삭제된 것으로 의심되는 경우 소프트웨어 레벨에서 방화벽에 의해 삭제된 모든 패킷의 카운터를 볼 수 있습니다.

```
<#root>
```

```
firepower#
```

```
show asp drop
```

```
Frame drop:
```

No route to host (no-route)	234
Flow is denied by configured rule (acl-drop)	71

```
Last clearing: 07:51:52 UTC Oct 10 2019 by enable_15
```

```
Flow drop:
```

```
Last clearing: 07:51:52 UTC Oct 10 2019 by enable_15
```


모든 ASP 소프트웨어 수준 삭제를 보려면 캡처를 활성화할 수 있습니다.

```
<#root>
```

```
firepower#
```

```
capture ASP type asp-drop all buffer 33554432 headers-only
```

---

 **팁:** 패킷 내용에 관심이 없는 경우 패킷 헤더만 캡처할 수 있습니다(header-only 옵션). 이렇게 하면 캡처 버퍼에서 훨씬 많은 패킷을 캡처할 수 있습니다. 또한 캡처 버퍼의 크기를 32MB(버퍼 옵션)까지 늘릴 수 있습니다(기본값은 500Kbytes). 마지막으로, FTD 버전 6.3에서처럼 file-size 옵션을 사용하면 최대 10GBytes의 캡처 파일을 구성할 수 있습니다. 이 경우 캡처 콘텐츠는 pcap 형식으로만 볼 수 있습니다.

---

캡처 내용을 확인하려면 필터를 사용하여 검색 범위를 좁힐 수 있습니다.

```
<#root>
```

```
firepower#
```

```
show capture ASP | include 10.10.1.100
```

```
18: 07:51:57.823672 192.168.0.100.12410 > 10.10.1.100.80: S 1870382552:1870382552(0) win 8192 <mss
19: 07:51:58.074291 192.168.0.100.12411 > 10.10.1.100.80: S 2006489005:2006489005(0) win 8192 <mss
26: 07:52:00.830370 192.168.0.100.12410 > 10.10.1.100.80: S 1870382552:1870382552(0) win 8192 <mss
29: 07:52:01.080394 192.168.0.100.12411 > 10.10.1.100.80: S 2006489005:2006489005(0) win 8192 <mss
45: 07:52:06.824282 192.168.0.100.12410 > 10.10.1.100.80: S 1870382552:1870382552(0) win 8192 <mss
46: 07:52:07.074230 192.168.0.100.12411 > 10.10.1.100.80: S 2006489005:2006489005(0) win 8192 <mss
```

이 경우 패킷은 이미 인터페이스 레벨에서 추적되므로 삭제 이유는 ASP 캡처에 언급되지 않습니다. 패킷은 한 곳에서만 추적할 수 있습니다(인그레스 인터페이스 또는 ASP 드롭). 이 경우 여러 ASP 삭제를 수행하고 특정 ASP 삭제 이유를 설정하는 것이 좋습니다. 권장 접근 방식은 다음과 같습니다.

1. 현재 ASP 삭제 카운터를 지웁니다.

```
<#root>
```

```
firepower#
```

```
clear asp drop
```

2. 방화벽을 통해 문제를 해결하는 흐름을 보냅니다(테스트 실행).

3. ASP 삭제 카운터를 다시 확인하고 증가되는 카운터를 적어 둡니다.

```
<#root>
```

```
firepower#
```

```

show asp drop

Frame drop:
  No route to host (
no-route
)
  Flow is denied by configured rule (
acl-drop
)
  71
234

```

4. 표시된 특정 삭제에 대해 ASP 캡처를 활성화합니다.

```

<#root>
firepower#
capture ASP_NO_ROUTE type asp-drop no-route
firepower#
capture ASP_ACL_DROP type asp-drop acl-drop

```

5. 방화벽을 통해 문제를 해결하는 흐름을 보냅니다(테스트 실행).

6. ASP 캡처를 확인합니다. 이 경우 패킷이 누락된 경로로 인해 삭제되었습니다.

```

<#root>
firepower#
show capture ASP_NO_ROUTE | include 192.168.0.100.*10.10.1.100

 93: 07:53:52.381663 192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss
 95: 07:53:52.632337 192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss
101: 07:53:55.375392 192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss
102: 07:53:55.626386 192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss
116: 07:54:01.376231 192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss
117: 07:54:01.626310 192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss

```

작업 5. FTD Lina 연결 테이블을 확인합니다.

패킷이 이그레스 인터페이스 'X'에 도달하는 경우가 있을 수 있지만 어떤 이유에서든 인터페이스 'Y'를 이그레스(egress)합니다. 방화벽 이그레스 인터페이스 결정은 다음 작동 순서를 기반으로 합니다.

1. 설정된 연결 조회
2. NAT(Network Address Translation) 조회 - UN-NAT(destination NAT) 단계가 PBR 및 경로 조

회보다 우선합니다.

3. PBR(Policy-Based Routing)

4. 라우팅 테이블 조회

FTD 연결 테이블을 확인하려면

```
<#root>
```

```
firepower#
```

```
show conn
```

```
2 in use, 4 most used
```

```
Inspect Snort:
```

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

```
TCP
```

```
DMZ
```

```
 10.10.1.100:
```

```
80
```

```
INSIDE
```

```
 192.168.0.100:
```

```
11694
```

```
, idle 0:00:01, bytes 0, flags
```

```
aA N1
```

```
TCP
```

```
DMZ
```

```
 10.10.1.100:80
```

```
INSIDE
```

```
 192.168.0.100:
```

```
11693
```

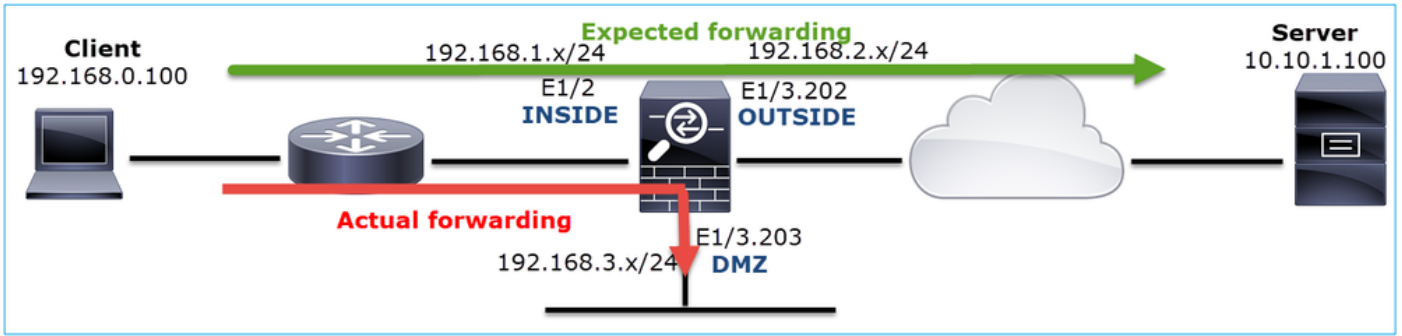
```
, idle 0:00:01, bytes 0, flags
```

```
aA N1
```

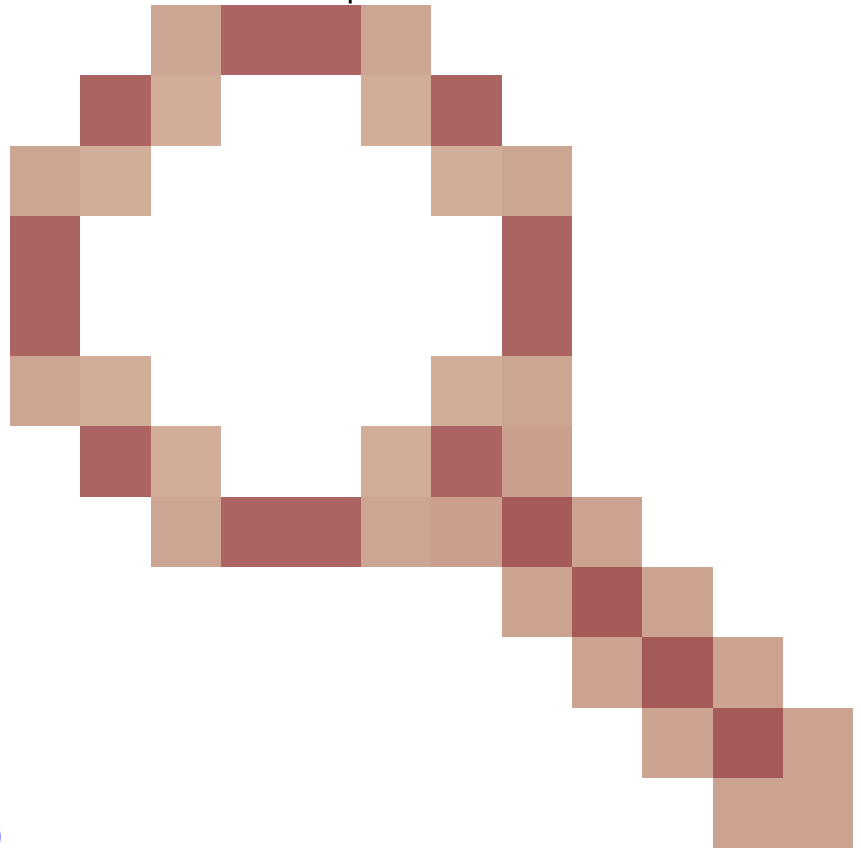
요점:

- 플래그(Aa)에 따라 연결이 원시(절반이 열림 - 방화벽에서 TCP SYN만 표시됨)입니다.
- 소스/목적지 포트에 따라 인그레스 인터페이스는 INSIDE이고 이그레스 인터페이스는 DMZ입니다.

다음 그림에서 시각화할 수 있습니다.



참고: 모든 FTD 인터페이스의 보안 레벨이 0이므로 show conn 출력의 인터페이스 순서는 인터페이스 번호를 기반으로 합니다. 구체적으로, vpif-num(virtual platform interface number)이 더 높은 인터페이스는 inside로 선택하고, vpif-num이 더 낮은 인터페이스는 outside로 선택합니다. show interface detail 명령을 사용하여 인터페이스 vpif 값을 볼 수 있습니다. 관련 개선



사항, Cisco 버그 ID [CSCvi15290](https://tools.cisco.com/bugcenter/bug/?bugID=CSCvi15290)  
 ENH: FTD는 FTD 'show conn' 출력에 연결 방향을 표시합니다.


```
<#root>
firepower#
show interface detail | i Interface number is|Interface [P|E].*is up
...
Interface Ethernet1/2 "INSIDE", is up, line protocol is up
  Interface number is
19
Interface Ethernet1/3.202 "OUTSIDE", is up, line protocol is up
  Interface number is
```

20

```
Interface Ethernet1/3.203 "DMZ", is up, line protocol is up
Interface number is
```

22

---

 참고: Firepower 소프트웨어 릴리스 6.5, ASA 릴리스 9.13.x에서 show conn long 및 show conn detail 명령 출력은 연결 개시자 및 응답자에 대한 정보를 제공합니다

---

출력 1:

```
<#root>
```

```
firepower#
```

```
show conn long
```

```
...
```

```
TCP OUTSIDE: 192.168.2.200/80 (192.168.2.200/80) INSIDE: 192.168.1.100/46050 (192.168.1.100/46050), flags
```

```
Initiator: 192.168.1.100, Responder: 192.168.2.200
```

```
Connection lookup keyid: 228982375
```

출력 2:

```
<#root>
```

```
firepower#
```

```
show conn detail
```

```
...
```

```
TCP OUTSIDE: 192.168.2.200/80 INSIDE: 192.168.1.100/46050,
flags aA N1, idle 4s, uptime 11s, timeout 30s, bytes 0
```

```
Initiator: 192.168.1.100, Responder: 192.168.2.200
```

```
Connection lookup keyid: 228982375
```

또한 show conn long은 Network Address Translation의 경우 괄호 안에 NATed IP를 표시합니다.

```
<#root>
```

```
firepower#
```

```
show conn long
```

```
...
```

```
TCP OUTSIDE: 192.168.2.222/80 (192.168.2.222/80) INSIDE: 192.168.1.100/34792 (192.168.2.150/34792), flags=0x00000000, win=0, len=0, seq=192.168.1.100, responder=192.168.2.222
Initiator: 192.168.1.100, Responder: 192.168.2.222
Connection lookup keyid: 262895
```

작업 6. 방화벽 ARP(Address Resolution Protocol) 캐시를 확인합니다.

방화벽에서 다음 홑을 확인할 수 없는 경우 방화벽은 원래 패킷(이 경우 TCP SYN)을 자동으로 삭제하고 다음 홑을 확인할 때까지 ARP 요청을 계속 전송합니다.

방화벽 ARP 캐시를 보려면 다음 명령을 사용합니다.

```
<#root>
firepower#
show arp
```

또한 확인되지 않은 호스트가 있는지 확인하려면 다음 명령을 사용할 수 있습니다.

```
<#root>
firepower#
show arp statistics
    Number of ARP entries in ASA: 0
    Dropped blocks in ARP: 84
    Maximum Queued blocks: 3
    Queued blocks: 0
    Interface collision ARPs Received: 0
    ARP-defense Gratuitous ARPs sent: 0
    Total ARP retries:
182          < indicates a possible issue for some hosts
    Unresolved hosts:
1
< this is the current status
    Maximum Unresolved hosts: 2
```

ARP 작업을 더 자세히 확인하려는 경우 ARP별 캡처를 활성화할 수 있습니다.

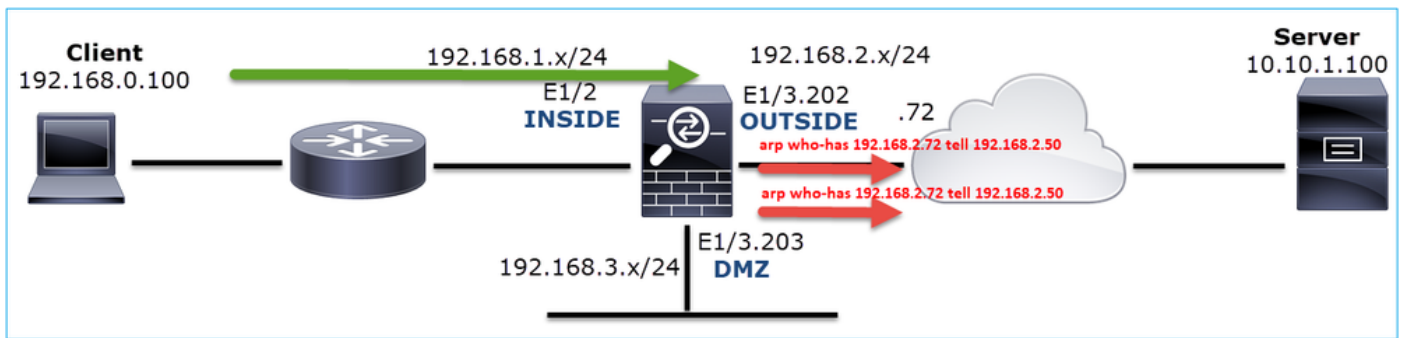
```
<#root>
```

```

firepower#
capture ARP ethernet-type arp interface OUTSIDE
firepower#
show capture ARP
...
 4: 07:15:16.877914      802.1Q vlan#202 P0 arp
who-has 192.168.2.72 tell 192.168.2.50
 5: 07:15:18.020033      802.1Q vlan#202 P0 arp who-has 192.168.2.72 tell 192.168.2.50

```

이 출력에서 방화벽(192.168.2.50)은 next-hop(192.168.2.72)을 확인하려고 하지만 ARP 응답이 없습니다



다음은 적절한 ARP 해결이 포함된 기능 시나리오의 출력입니다.

```

<#root>
firepower#
show capture ARP

2 packets captured

 1: 07:17:19.495595      802.1Q vlan#202 P0
arp who-has 192.168.2.72 tell 192.168.2.50
 2: 07:17:19.495946      802.1Q vlan#202 P0
arp reply 192.168.2.72 is-at 4c:4e:35:fc:fc:d8
2 packets shown

```

```

<#root>
firepower#
show arp

INSIDE 192.168.1.71 4c4e.35fc.fcd8 9
OUTSIDE 192.168.2.72 4c4e.35fc.fcd8 9

```



ARP 항목이 없는 경우 라이브 TCP SYN 패킷의 추적은 다음과 같습니다.

```
<#root>
```

```
firepower#
```

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

```
6 packets captured
```

```
1: 07:03:43.270585
```

```
192.168.0.100.11997 > 10.10.1.100.80
```

```
: S 4023707145:4023707145(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
```

```
Phase: 1
```

```
Type: CAPTURE
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
MAC Access list
```

```
Phase: 2
```

```
Type: ACCESS-LIST
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Implicit Rule
```

```
Additional Information:
```

```
MAC Access list
```

```
Phase: 3
```

```
Type: ROUTE-LOOKUP
```

```
Subtype: Resolve Egress Interface
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
found next-hop 192.168.2.72 using egress ifc OUTSIDE
```

```
...
```

```
Phase: 14
```

```
Type: FLOW-CREATION
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
New flow created with id 4814, packet dispatched to next module
```

```
...
```

```
Phase: 17
```

```
Type: ROUTE-LOOKUP
```

```
Subtype: Resolve Egress Interface
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
found next-hop 192.168.2.72 using egress ifc OUTSIDE
```

```
Result:
```

```
input-interface: INSIDE
```

```
input-status: up
```

```
input-line-status: up
output-interface: OUTSIDE
output-status: up
output-line-status: up
Action: allow
```

출력에서 볼 수 있는 것처럼 다음 hops에 연결할 수 없고 패킷이 방화벽에 의해 자동으로 삭제되는 경우에도 Action: allow(작업: 허용)가 추적에 표시됩니다. 이 경우 패킷 추적기 툴은 보다 정확한 출력을 제공하므로 반드시 확인해야 합니다.

<#root>

```
firepower#
```

```
packet-tracer input INSIDE tcp 192.168.0.100 1111 10.10.1.100 80
```

```
Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
MAC Access list
```

```
Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
MAC Access list
```

```
Phase: 3
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 192.168.2.72 using egress ifc OUTSIDE
```

```
...
```

```
Phase: 14
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 4816, packet dispatched to next module
```

```
...
```

```
Phase: 17
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
```

found next-hop 192.168.2.72 using egress ifc OUTSIDE

Result:

input-interface: INSIDE  
input-status: up  
input-line-status: up  
output-interface: OUTSIDE  
output-status: up  
output-line-status: up  
Action: drop

Drop-reason: (no-v4-adjacency) No valid V4 adjacency, Drop-location: frame 0x00005647a4e86109 flow (NA),

최근 ASA/Firepower 버전에서는 이전 메시지가 다음으로 최적화되었습니다.

<#root>

Drop-reason: (no-v4-adjacency) No valid V4 adjacency.

Check ARP table (show arp) has entry for nexthop

., Drop-location: f

### 가능한 원인 및 권장 조치 요약

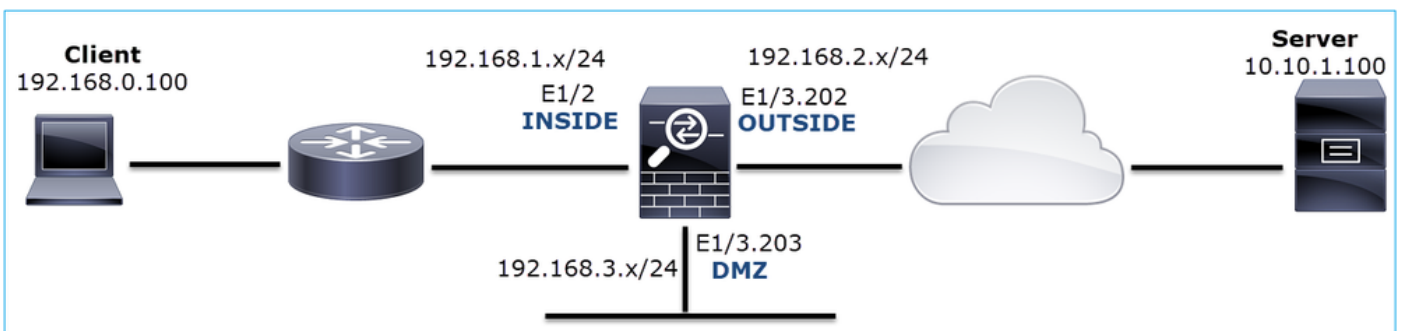
인그레스 인터페이스의 TCP SYN 패킷만 표시되지만 예상 이그레스 인터페이스에서 전송된 TCP SYN 패킷이 없는 경우 다음과 같은 원인이 있을 수 있습니다.

가능한 원인	권장 작업
패킷은 방화벽 액세스 정책에 의해 삭제됩니다.	<ul style="list-style-type: none"> <li>• 방화벽이 패킷을 처리하는 방법을 보려면 패킷 추적기 또는 캡처 w/trace를 사용합니다.</li> <li>• 방화벽 로그를 확인합니다.</li> <li>• 방화벽 ASP 삭제를 확인합니다(show asp drop or capture type asp-drop).</li> <li>• FMC Connection Events를 선택합니다. 이 경우 규칙에 로깅이 활성화되어 있다고 가정합니다.</li> </ul>
캡처 필터가 잘못되었습니다.	<ul style="list-style-type: none"> <li>• 소스 또는 목적지 IP를 수정하는 NAT 변환이 있는지 확인하려면 패킷 추적기 또는 캡처 w/trace를 사용합니다. 이 경우 캡처 필터를 조정합니다.</li> <li>• show conn long 명령 출력에서는 NATed IP를 보여줍니다.</li> </ul>

<p>패킷은 다른 이그레스 인터페이스로 전송됩니다.</p>	<ul style="list-style-type: none"> <li>• 방화벽이 패킷을 처리하는 방법을 보려면 packet-tracer 또는 capture w/trace를 사용합니다. 이그레스 인터페이스 결정, 현재 연결, UN-NAT, PBR 및 라우팅 테이블 조회를 고려하는 작업의 순서를 기억하십시오.</li> <li>• 방화벽 로그를 확인합니다.</li> <li>• 방화벽 연결 테이블(show conn)을 확인합니다.</li> </ul> <p>패킷이 현재 연결과 일치하기 때문에 잘못된 인터페이스로 전송되는 경우 clear conn address 명령을 사용하고 지울 연결의 5-tuple을 지정합니다.</p>
<p>목적지까지 가는 길이 없어요</p>	<ul style="list-style-type: none"> <li>• 방화벽이 패킷을 처리하는 방법을 보려면 패킷 추적기 또는 캡처 w/trace를 사용합니다.</li> <li>• 방화벽 ASP 삭제(show asp drop)에서 no-route 삭제 이유를 확인합니다.</li> </ul>
<p>이그레스 인터페이스에 ARP 항목이 없습니다.</p>	<ul style="list-style-type: none"> <li>• 방화벽 ARP 캐시를 확인합니다(show arp).</li> <li>• packet-tracer를 사용하여 유효한 인접성이 있는지 확인합니다.</li> </ul>
<p>이그레스 인터페이스가 다운되었습니다.</p>	<p>방화벽에서 show interface ip brief 명령의 출력을 확인하고 인터페이스 상태를 확인합니다.</p>

## 사례 2. 클라이언트의 TCP SYN, 서버의 TCP RST

이 그림에서는 토폴로지를 보여줍니다.



문제 설명: HTTP가 작동하지 않음

영향을 받는 흐름:

소스 IP: 192.168.0.100

Dst IP: 10.10.1.100

프로토콜: TCP 80

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

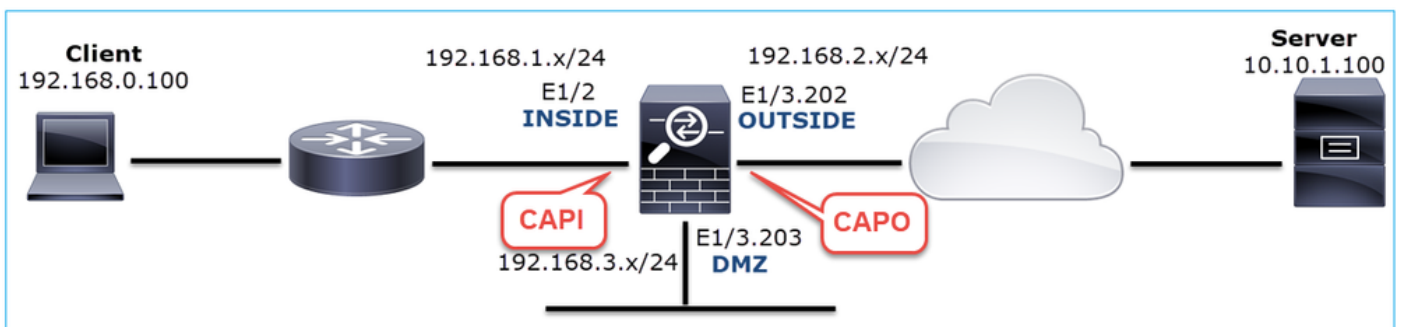
```
<#root>
```

```
firepower#
```

```
capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100
```

```
firepower#
```

```
capture CAPO int OUTSIDE match ip host 192.168.0.100 host 10.10.1.100
```



캡처 - 작동하지 않는 시나리오:

디바이스 CLI에서 캡처는 다음과 같이 표시됩니다.

```
<#root>
```

```
firepower#
```

```
show capture
```

```
capture CAPI type raw-data trace interface INSIDE [Capturing -
```

```
834 bytes
```

```
]
```

```
match ip host 192.168.0.100 host 10.10.1.100
```

```
capture CAPO type raw-data interface OUTSIDE [Capturing -
```

```
878 bytes
```

```
]
```

```
match ip host 192.168.0.100 host 10.10.1.100
```

## CAPI 내용:

<#root>

firepower#

show capture CAPI

```
1: 05:20:36.654217 192.168.0.100.22195 > 10.10.1.100.80:
S
1397289928:1397289928(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
2: 05:20:36.904311 192.168.0.100.22196 > 10.10.1.100.80:
S
2171673258:2171673258(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
3: 05:20:36.905043 10.10.1.100.80 > 192.168.0.100.22196:
R
1850052503:1850052503(0) ack 2171673259 win 0
4: 05:20:37.414132 192.168.0.100.22196 > 10.10.1.100.80:
S
2171673258:2171673258(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
5: 05:20:37.414803 10.10.1.100.80 > 192.168.0.100.22196:
R
31997177:31997177(0) ack 2171673259 win 0
6: 05:20:37.914183 192.168.0.100.22196 > 10.10.1.100.80:
S
2171673258:2171673258(0) win 8192 <mss 1460,nop,nop,sackOK>
...
```

## CAPO 내용:

<#root>

firepower#

show capture CAPO

```
1: 05:20:36.654507 802.1Q vlan#202 P0 192.168.0.100.22195 > 10.10.1.100.80:
S
2866789268:2866789268(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>
2: 05:20:36.904478 802.1Q vlan#202 P0 192.168.0.100.22196 > 10.10.1.100.80:
S
4785344:4785344(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>
3: 05:20:36.904997 802.1Q vlan#202 P0 10.10.1.100.80 > 192.168.0.100.22196:
R
```

```
0:0(0) ack 4785345 win 0
4: 05:20:37.414269 802.1Q vlan#202 PO 192.168.0.100.22196 > 10.10.1.100.80:
```

S

```
4235354730:4235354730(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>
5: 05:20:37.414758 802.1Q vlan#202 PO 10.10.1.100.80 > 192.168.0.100.22196:
```

R

```
0:0(0) ack 4235354731 win 0
6: 05:20:37.914305 802.1Q vlan#202 PO 192.168.0.100.22196 > 10.10.1.100.80:
```

S

```
4118617832:4118617832(0) win 8192 <mss 1380,nop,nop,sackOK>
```

이 이미지는 Wireshark에서 CAPI를 캡처한 것을 보여줍니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.100	10.10.1.100	TCP	66	22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.250094	192.168.0.100	10.10.1.100	TCP	66	22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	0.000732	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	0.509089	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
5	0.000671	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=2476911971 Ack=1 Win=0 Len=0
6	0.499380	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
7	0.000625	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=2853655305 Ack=1 Win=0 Len=0
8	1.739729	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
9	0.000611	10.10.1.100	192.168.0.100	TCP	54	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10	0.499385	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
11	0.000671	10.10.1.100	192.168.0.100	TCP	54	80 → 22195 [RST, ACK] Seq=151733665 Ack=1 Win=0 Len=0

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
 > Ethernet II, Src: Cisco\_fc:fc:d8 (4c:4e:35:fc:d8), Dst: Cisco\_f6:1d:ae (00:be:75:f6:1d:ae)  
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100  
 > Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Seq: 0, Len: 0

요점:

1. 소스가 TCP SYN 패킷을 전송합니다.
2. TCP RST는 소스로 전송됩니다.
3. 소스가 TCP SYN 패킷을 재전송합니다.
4. MAC 주소가 정확합니다(소스 MAC 주소가 다운스트림 라우터에 속하는 인그레스 패킷에서 는 대상 MAC 주소가 방화벽 INSIDE 인터페이스에 속함).

이 이미지는 Wireshark에서 CAPO를 캡처한 것을 보여줍니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-11 07:20:36.654507	192.168.0.100	10.10.1.100	TCP	70	22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
2	2019-10-11 07:20:36.904478	192.168.0.100	10.10.1.100	TCP	70	22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
3	2019-10-11 07:20:36.904997	10.10.1.100	192.168.0.100	TCP	58	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	2019-10-11 07:20:37.414269	192.168.0.100	10.10.1.100	TCP	70	[TCP Port numbers reused] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
5	2019-10-11 07:20:37.414758	10.10.1.100	192.168.0.100	TCP	58	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6	2019-10-11 07:20:37.914305	192.168.0.100	10.10.1.100	TCP	66	[TCP Port numbers reused] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 SACK_PERM=1
7	2019-10-11 07:20:37.914762	10.10.1.100	192.168.0.100	TCP	58	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8	2019-10-11 07:20:39.654629	192.168.0.100	10.10.1.100	TCP	70	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
9	2019-10-11 07:20:39.655102	10.10.1.100	192.168.0.100	TCP	58	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10	2019-10-11 07:20:40.154700	192.168.0.100	10.10.1.100	TCP	66	[TCP Port numbers reused] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 SACK_PERM=1
11	2019-10-11 07:20:40.155173	10.10.1.100	192.168.0.100	TCP	58	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)  
 > Ethernet II, Src: Cisco\_f6:1d:8e (00:be:75:f6:1d:8e), Dst: Cisco\_fc:fc:d8 (4c:4e:35:fc:d8)  
 > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202  
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100  
 > Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Seq: 0, Len: 0

요점:

1. 소스가 TCP SYN 패킷을 전송합니다.
2. TCP RST는 OUTSIDE 인터페이스에 도착합니다.
3. 소스가 TCP SYN 패킷을 재전송합니다.
4. MAC 주소가 정확합니다(이그레스 패킷에서는 방화벽 OUTSIDE가 소스 MAC이고 업스트림 라우터가 대상 MAC임).

2개의 캡처를 바탕으로 다음과 같은 결론을 내릴 수 있습니다.

- 클라이언트와 서버 간의 TCP 3-way 핸드셰이크가 완료되지 않습니다
- 방화벽 이그레스 인터페이스에 도착하는 TCP RST가 있습니다
- 방화벽이 적절한 업스트림 및 다운스트림 디바이스에 '연결'(MAC 주소 기반)

## 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. TCP RST를 전송하는 소스 MAC 주소를 확인합니다.

TCP SYN 패킷에 표시된 대상 MAC이 TCP RST 패킷에 표시된 소스 MAC과 동일한지 확인합니다.

The image displays two screenshots from Wireshark, labeled 'CAPO\_RST\_SERVER.pcap'. The top screenshot shows a list of captured packets. Packet 2 is a TCP SYN packet from source IP 192.168.0.100 to destination IP 10.10.1.100. The packet details pane shows the Ethernet II header with Source MAC: Cisco\_f6:1d:8e (00:be:75:f6:1d:8e) and Destination MAC: Cisco\_fc:fc:d8 (4c:4e:35:fc:fc:d8). The bottom screenshot shows packet 3, a TCP RST, ACK packet from source IP 10.10.1.100 to destination IP 192.168.0.100. The packet details pane shows the Ethernet II header with Source MAC: Cisco\_fc:fc:d8 (4c:4e:35:fc:fc:d8) and Destination MAC: Cisco\_f6:1d:8e (00:be:75:f6:1d:8e). Two arrows, one orange and one green, cross between the two screenshots, pointing from the source MAC of the SYN packet to the destination MAC of the RST packet, and from the source MAC of the RST packet to the destination MAC of the SYN packet, illustrating the verification process.

이 확인은 다음 2가지를 확인하는 데 목적이 있습니다.

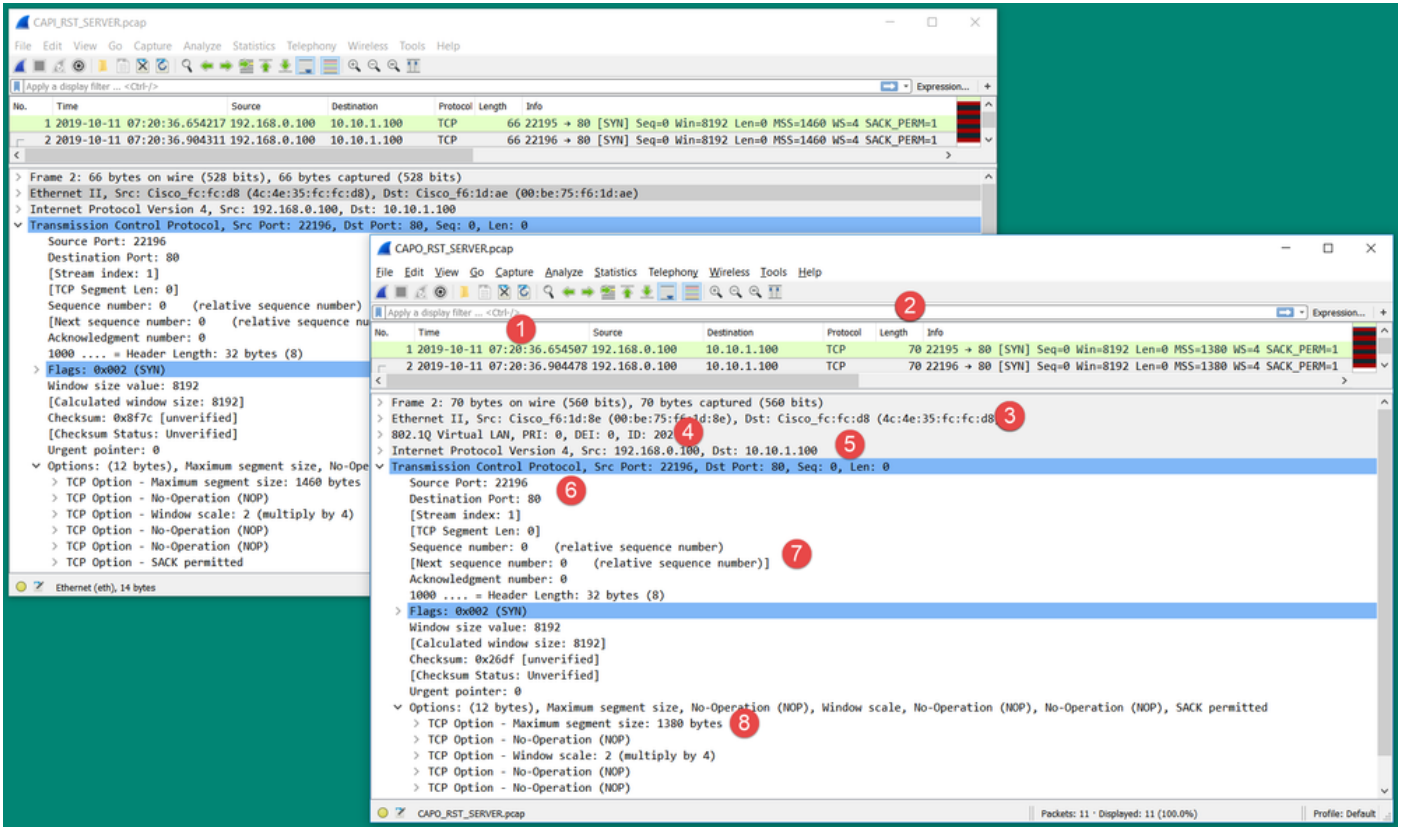
- 비대칭 흐름이 없는지 확인합니다.
- MAC가 예상 업스트림 디바이스에 속하는지 확인합니다.

작업 2. 인그레스 패킷과 이그레스 패킷을 비교합니다.

Wireshark에서 2개의 패킷을 시각적으로 비교하여 방화벽에서 패킷을 수정/손상시키지 않는지 확



인합니다. 예상되는 몇 가지 차이점이 강조 표시됩니다.



### 요점:

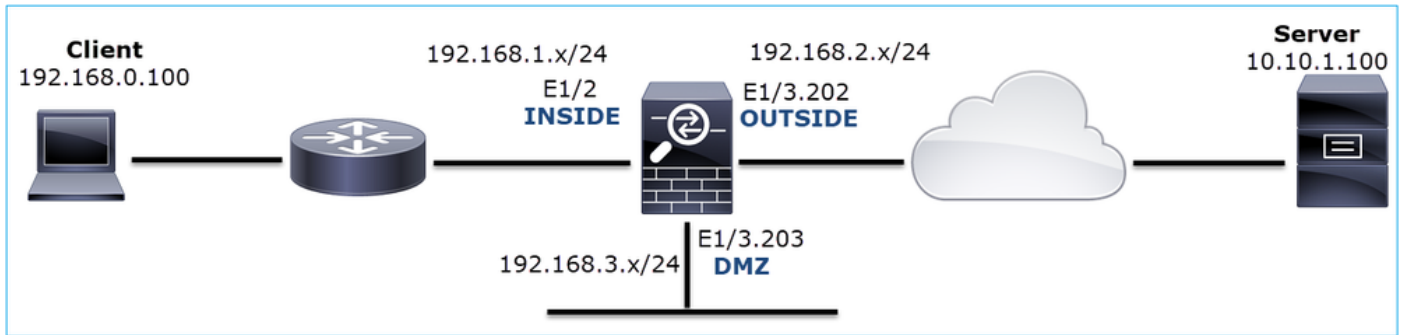
1. 타임스탬프는 다릅니다. 반면 그 차이는 작고 합리적이어서야 한다. 이는 패킷에 적용된 기능 및 정책 검사와 디바이스의 로드 여부에 따라 달라집니다.
2. 패킷의 길이는 특히 한쪽 면에만 방화벽에 의해 추가/제거된 dot1Q 헤더가 있는 경우 달라질 수 있습니다.
3. MAC 주소가 다릅니다.
4. 하위 인터페이스에서 캡처를 수행한 경우 dot1Q 헤더를 사용할 수 있습니다.
5. NAT 또는 PAT(Port Address Translation)가 패킷에 적용되는 경우 IP 주소가 달라집니다.
6. NAT 또는 PAT가 패킷에 적용되는 경우 소스 또는 목적지 포트가 다릅니다.
7. Wireshark Relative Sequence Number 옵션을 비활성화하면 ISN(Initial Sequence Number) 임의의 설정으로 인해 방화벽에 의해 TCP 시퀀스 번호/승인 번호가 수정됩니다.
8. 일부 TCP 옵션은 덮어쓸 수 있습니다. 예를 들어, 트랜짓 경로의 패킷 단편화를 방지하기 위해 방화벽은 기본적으로 TCP MSS(Maximum Segment Size)를 1380으로 변경합니다.

### 작업 3. 목적지에서 캡처합니다.

가능하면 목적지에서 캡처합니다. 이것이 가능하지 않으면 최대한 목적지에 가까운 곳에 캡처를 취하십시오. 여기서 목표는 누가 TCP RST를 전송하는지 확인하는 것입니다(대상 서버 또는 경로에 있는 다른 디바이스?).

### 사례 3. TCP 3-Way 핸드셰이크 + 한 엔드포인트의 RST

이 그림에서는 토폴로지를 보여줍니다.



문제 설명: HTTP가 작동하지 않음

영향을 받는 흐름:

소스 IP: 192.168.0.100

Dst IP: 10.10.1.100

프로토콜: TCP 80

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

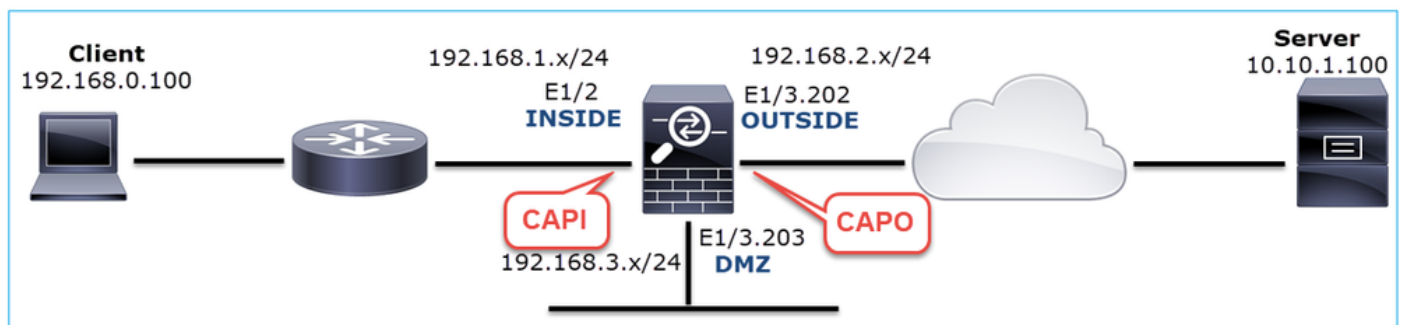
<#root>

firepower#

```
capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100
```

firepower#

```
capture CAPO int OUTSIDE match ip host 192.168.0.100 host 10.10.1.100
```



캡처 - 작동하지 않는 시나리오:

이 문제가 캡처에서 나타날 수 있는 몇 가지 방법이 있습니다.

### 3.1 - TCP 3-way 핸드셰이크 + 클라이언트에서 지연된 RST

방화벽은 CAPI를 캡처하고 CAPO는 이미지에 표시된 것과 같은 패킷을 포함합니다.

No.	Time	Source	Destination	Protocol	Length	Info
2	2019-10-13 17:06:27.874085	192.168.0.100	10.10.1.100	TCP	66	48295 → 80 [SYN] Seq=179631561 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	2019-10-13 17:06:27.874741	10.10.1.100	192.168.0.100	TCP	66	80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
4	2019-10-13 17:06:27.875183	192.168.0.100	10.10.1.100	TCP	54	48295 → 80 [ACK] Seq=179631562 Ack=3838911938 Win=66240 Len=0
8	2019-10-13 17:06:30.882537	10.10.1.100	192.168.0.100	TCP	66	[TCP Retransmission] 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
9	2019-10-13 17:06:30.883056	192.168.0.100	10.10.1.100	TCP	66	[TCP Previous segment not captured] 48295 → 80 [ACK] Seq=179631962 Ack=3838911938 Win=66240 Len=0 SLE=3838911937 SRE=3838911938
13	2019-10-13 17:06:36.889022	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=65535 Len=0 MSS=1380 SACK_PERM=1
14	2019-10-13 17:06:36.889526	192.168.0.100	10.10.1.100	TCP	66	[TCP Dup ACK 4#1] 48295 → 80 [ACK] Seq=179631962 Ack=3838911938 Win=66240 Len=0 SLE=3838911937 SRE=3838911938
17	2019-10-13 17:06:47.943631	192.168.0.100	10.10.1.100	TCP	54	48295 → 80 [RST, ACK] Seq=179631962 Ack=3838911938 Win=0 Len=0

요점:

1. TCP 3-way 핸드셰이크는 방화벽을 통과합니다.
2. 서버가 SYN/ACK를 다시 전송합니다.
3. 클라이언트가 ACK를 재전송합니다.
4. ~20초 후 클라이언트는 TCP RST를 포기하고 전송합니다.

권장 작업

이 섹션에 나와 있는 조치는 문제를 더 줄이기 위한 목적입니다.

작업 1. 가능한 한 두 엔드포인트 가까이에 캡처를 배치합니다.

방화벽 캡처는 클라이언트 ACK가 서버에서 처리되지 않았음을 나타냅니다. 이는 다음과 같은 사실을 기반으로 합니다.

- 서버가 SYN/ACK를 다시 전송합니다.
- 클라이언트가 ACK를 재전송합니다.
- 클라이언트는 데이터 이전에 TCP RST 또는 FIN/ACK를 전송합니다.

서버에서 캡처하면 문제가 표시됩니다. TCP 3-way 핸드셰이크의 클라이언트 ACK가 도착하지 않았습니다.

26	7.636612	192.168.0.100	10.10.1.100	TCP	66	55324→80 [SYN] Seq=433201323 Win=8192 Len=0 MSS=1380 WS=4 SAC...
29	7.637571	10.10.1.100	192.168.0.100	TCP	66	80→55324 [SYN, ACK] Seq=4063222169 Ack=433201324 Win=8192 Len...
30	7.930152	192.168.0.100	10.10.1.100	TCP	66	55325→80 [SYN] Seq=366197499 Win=8192 Len=0 MSS=1380 WS=4 SAC...
31	7.930221	10.10.1.100	192.168.0.100	TCP	66	80→55325 [SYN, ACK] Seq=2154790336 Ack=366197500 Win=8192 Len...
41	10.629868	192.168.0.100	10.10.1.100	TCP	66	[TCP Spurious Retransmission] 55324→80 [SYN] Seq=433201323 Wi...
42	10.633208	10.10.1.100	192.168.0.100	TCP	66	[TCP Retransmission] 80→55324 [SYN, ACK] Seq=4063222169 Ack=4...
44	10.945178	10.10.1.100	192.168.0.100	TCP	66	[TCP Retransmission] 80→55325 [SYN, ACK] Seq=2154790336 Ack=3...
60	16.636255	192.168.0.100	10.10.1.100	TCP	62	[TCP Spurious Retransmission] 55324→80 [SYN] Seq=433201323 Wi...
61	16.639145	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80→55324 [SYN, ACK] Seq=4063222169 Ack=4...
62	16.951195	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80→55325 [SYN, ACK] Seq=2154790336 Ack=3...

### 3.2 - TCP 3-way 핸드셰이크 + 클라이언트에서 지연된 FIN/ACK + 서버에서 지연된 RST

방화벽은 CAPI를 캡처하고 CAPO는 이미지에 표시된 것과 같은 패킷을 포함합니다.

25	2019-10-13 17:07:06.853334	192.168.0.100	10.10.1.100	TCP	66	48299 → 80 [SYN] Seq=3239914002 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
29	2019-10-13 17:07:09.852922	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 48299 → 80 [SYN] Seq=3239914002 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
30	2019-10-13 17:07:09.854844	10.10.1.100	192.168.0.100	TCP	66	80 → 48299 [SYN, ACK] Seq=808763519 Ack=3239914003 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
31	2019-10-13 17:07:09.855287	192.168.0.100	10.10.1.100	TCP	54	48299 → 80 [ACK] Seq=3239914003 Ack=808763520 Win=66240 Len=0
34	2019-10-13 17:07:14.856996	192.168.0.100	10.10.1.100	TCP	54	48299 → 80 [FIN, ACK] Seq=3239914003 Ack=808763520 Win=66240 Len=0
35	2019-10-13 17:07:15.861451	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80 → 48299 [SYN, ACK] Seq=808763519 Ack=3239914003 Win=65535 Len=0 MSS=1380 SACK_PERM=1
36	2019-10-13 17:07:15.861970	192.168.0.100	10.10.1.100	TCP	66	[TCP Dup ACK 31#1] 48299 → 80 [ACK] Seq=3239914004 Ack=808763520 Win=66240 Len=0 SLE=808763519 SRE=808763520
39	2019-10-13 17:07:17.854051	192.168.0.100	10.10.1.100	TCP	54	[TCP Retransmission] 48299 → 80 [FIN, ACK] Seq=3239914003 Ack=808763520 Win=66240 Len=0
40	2019-10-13 17:07:23.855012	192.168.0.100	10.10.1.100	TCP	54	[TCP Retransmission] 48299 → 80 [FIN, ACK] Seq=3239914003 Ack=808763520 Win=66240 Len=0
46	2019-10-13 17:07:27.858949	10.10.1.100	192.168.0.100	TCP	54	80 → 48299 [RST] Seq=808763520 Win=0 Len=0

요점:

1. TCP 3-way 핸드셰이크는 방화벽을 통과합니다.
2. ~5초 후에 클라이언트가 FIN/ACK을 전송합니다.
3. ~20초 후에 서버가 중단되고 TCP RST를 전송합니다.

이 캡처에 따라 방화벽을 통과하는 TCP 3-way 핸드셰이크가 있지만 실제로 하나의 엔드포인트에서 완료되지 않는 것 같다고 결론을 내릴 수 있습니다(재전송 시 이를 나타냄).

권장 작업

case 3.1과 동일

3.3 - TCP 3-way 핸드셰이크 + 클라이언트에서 지연된 RST

방화벽은 CAPI를 캡처하고 CAPO는 이미지에 표시된 것과 같은 패킷을 포함합니다.

No.	Time	Source	Destination	Protocol	Length	Info
129	2019-10-13 17:09:20.513355	192.168.0.100	10.10.1.100	TCP	66	48355 → 80 [SYN] Seq=2581697538 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
130	2019-10-13 17:09:20.514011	10.10.1.100	192.168.0.100	TCP	66	80 → 48355 [SYN, ACK] Seq=1633018698 Ack=2581697539 Win=8192 Len=0 MSS=1460
131	2019-10-13 17:09:20.514438	192.168.0.100	10.10.1.100	TCP	54	48355 → 80 [ACK] Seq=2581697539 Ack=1633018699 Win=66240 Len=0
132	2019-10-13 17:09:39.473089	192.168.0.100	10.10.1.100	TCP	54	48355 → 80 [RST, ACK] Seq=2581697939 Ack=1633018699 Win=0 Len=0

요점:

1. TCP 3-way 핸드셰이크는 방화벽을 통과합니다.
2. ~20초 후 클라이언트는 TCP RST를 포기하고 전송합니다.

이러한 캡처를 바탕으로 다음과 같은 결론을 내릴 수 있습니다.

- 5-20초 후에 한 엔드포인트가 연결을 종료하고 연결을 종료하기로 결정합니다.

권장 작업

case 3.1과 동일

3.4 - TCP 3-way 핸드셰이크 + 서버에서 즉시 RST

그림과 같이 두 방화벽 모두 CAPI와 CAPO에 이러한 패킷을 포함합니다.

No.	Time	Source	Destination	Protocol	Length	Info
26	2019-10-13 17:07:07.104410	192.168.0.100	10.10.1.100	TCP	66	48300 → 80 [SYN] Seq=2563435279 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
27	2019-10-13 17:07:07.105112	10.10.1.100	192.168.0.100	TCP	66	80 → 48300 [SYN, ACK] Seq=3757137497 Ack=2563435280 Win=8192 Len=0 MSS=1380
28	2019-10-13 17:07:07.105554	192.168.0.100	10.10.1.100	TCP	54	48300 → 80 [ACK] Seq=2563435280 Ack=3757137498 Win=66240 Len=0
41	2019-10-13 17:07:07.106325	10.10.1.100	192.168.0.100	TCP	54	80 → 48300 [RST] Seq=2563435280 Win=0 Len=0

요점:

1. TCP 3-way 핸드셰이크는 방화벽을 통과합니다.
2. ACK 패킷 이후 몇 밀리초 후에 서버의 TCP RST가 있습니다.

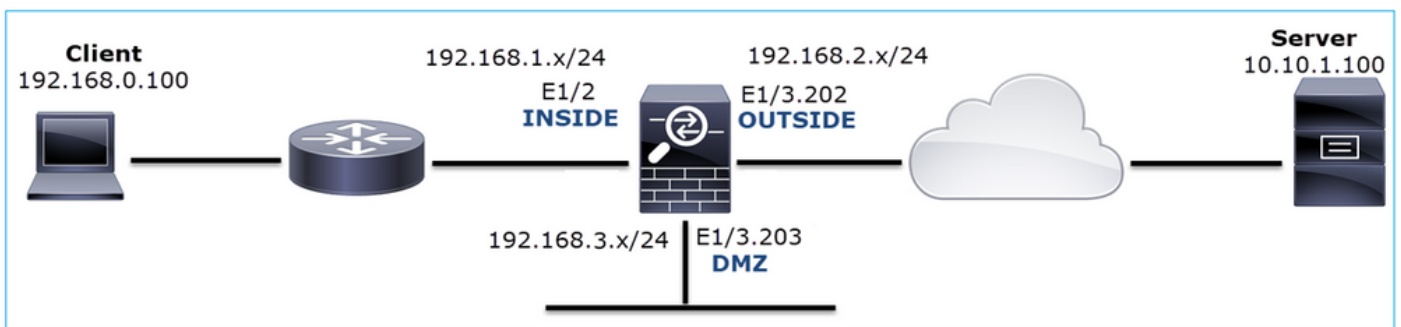
## 권장 작업

조치: 캡처를 가능한 한 서버에 가깝게 수행합니다.

서버의 즉각적인 TCP RST는 작동하지 않는 서버 또는 TCP RST를 전송하는 경로에 있는 디바이스를 나타낼 수 있습니다. 서버 자체에서 캡처를 수행하고 TCP RST의 소스를 확인합니다.

## 사례 4. 클라이언트의 TCP RST

이 그림에서는 토폴로지를 보여줍니다.



문제 설명: HTTP가 작동하지 않습니다.

영향을 받는 흐름:

소스 IP: 192.168.0.100

Dst IP: 10.10.1.100

프로토콜: TCP 80

## 캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

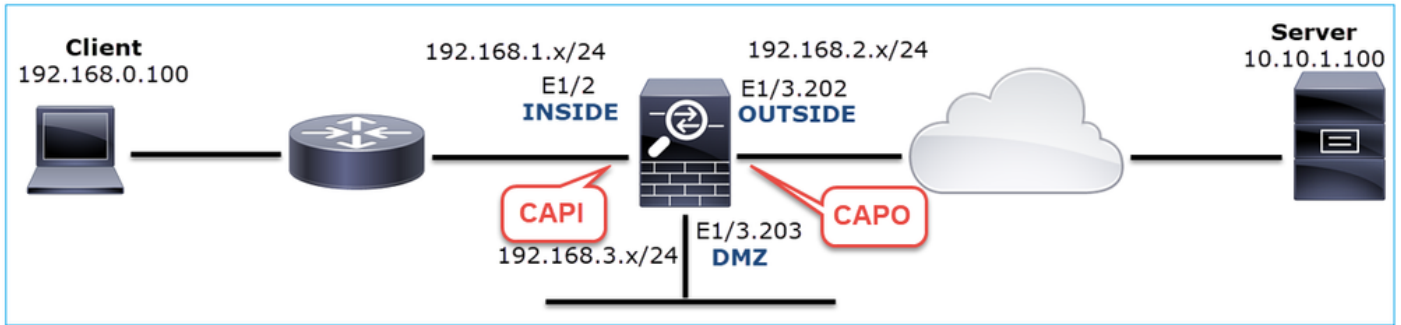
```
<#root>
```

```
firepower#
```

```
capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100
```

```
firepower#
```

```
capture CAPO int OUTSIDE match ip host 192.168.0.100 host 10.10.1.100
```



캡처 - 작동하지 않는 시나리오:

CAPI 내용입니다.

```
<#root>
```

```
firepower#
```

```
show capture CAPI
```

14 packets captured

```

1: 12:32:22.860627 192.168.0.100.47078 > 10.10.1.100.80: S 4098574664:4098574664(0) win 8192 <mss
2: 12:32:23.111307 192.168.0.100.47079 > 10.10.1.100.80: S 2486945841:2486945841(0) win 8192 <mss
3: 12:32:23.112390 192.168.0.100.47079 > 10.10.1.100.80: R 3000518858:3000518858(0) win 0
4: 12:32:25.858109 192.168.0.100.47078 > 10.10.1.100.80: S 4098574664:4098574664(0) win 8192 <mss
5: 12:32:25.868698 192.168.0.100.47078 > 10.10.1.100.80: R 1386249853:1386249853(0) win 0
6: 12:32:26.108118 192.168.0.100.47079 > 10.10.1.100.80: S 2486945841:2486945841(0) win 8192 <mss
7: 12:32:26.109079 192.168.0.100.47079 > 10.10.1.100.80: R 3000518858:3000518858(0) win 0
8: 12:32:26.118295 192.168.0.100.47079 > 10.10.1.100.80: R 3000518858:3000518858(0) win 0
9: 12:32:31.859925 192.168.0.100.47078 > 10.10.1.100.80: S 4098574664:4098574664(0) win 8192 <mss
10: 12:32:31.860902 192.168.0.100.47078 > 10.10.1.100.80: R 1386249853:1386249853(0) win 0
11: 12:32:31.875229 192.168.0.100.47078 > 10.10.1.100.80: R 1386249853:1386249853(0) win 0
12: 12:32:32.140632 192.168.0.100.47079 > 10.10.1.100.80: R 3000518858:3000518858(0) win 0
13: 12:32:32.159995 192.168.0.100.47079 > 10.10.1.100.80: S 2486945841:2486945841(0) win 8192 <mss
14: 12:32:32.160956 192.168.0.100.47079 > 10.10.1.100.80: R 3000518858:3000518858(0) win 0

```

14 packets shown

다음은 CAPO 내용입니다.

```
<#root>
```

```
firepower#
```

```
show capture CAPO
```

11 packets captured

```

1: 12:32:22.860780 802.1Q vlan#202 PO 192.168.0.100.47078 > 10.10.1.100.80: S 1386249852:138624985
2: 12:32:23.111429 802.1Q vlan#202 PO 192.168.0.100.47079 > 10.10.1.100.80: S 3000518857:300051885
3: 12:32:23.112405 802.1Q vlan#202 PO 192.168.0.100.47079 > 10.10.1.100.80: R 3514091874:351409187
4: 12:32:25.858125 802.1Q vlan#202 PO 192.168.0.100.47078 > 10.10.1.100.80: S 1386249852:138624985
5: 12:32:25.868729 802.1Q vlan#202 PO 192.168.0.100.47078 > 10.10.1.100.80: R 2968892337:296889233
6: 12:32:26.108240 802.1Q vlan#202 PO 192.168.0.100.47079 > 10.10.1.100.80: S 3822259745:382225974

```

```

7: 12:32:26.109094 802.1Q vlan#202 PO 192.168.0.100.47079 > 10.10.1.100.80: R 40865466:40865466(0)
8: 12:32:31.860062 802.1Q vlan#202 PO 192.168.0.100.47078 > 10.10.1.100.80: S 4294058752:429405875
9: 12:32:31.860917 802.1Q vlan#202 PO 192.168.0.100.47078 > 10.10.1.100.80: R 1581733941:158173394
10: 12:32:32.160102 802.1Q vlan#202 PO 192.168.0.100.47079 > 10.10.1.100.80: S 4284301197:428430119
11: 12:32:32.160971 802.1Q vlan#202 PO 192.168.0.100.47079 > 10.10.1.100.80: R 502906918:502906918(
11 packets shown

```

방화벽 로그에 표시되는 내용은 다음과 같습니다.

```
<#root>
```

```
firepower#
```

```
show log | i 47741
```

```

Oct 13 2019 13:57:36: %FTD-6-302013: Built inbound TCP connection 4869 for INSIDE:192.168.0.100/47741 (
Oct 13 2019 13:57:36: %FTD-6-302014: Teardown TCP connection 4869 for INSIDE:192.168.0.100/47741 to OUT

```

```
TCP Reset-O from INSIDE
```

```

Oct 13 2019 13:57:39: %FTD-6-302013: Built inbound TCP connection 4870 for INSIDE:192.168.0.100/47741 (
Oct 13 2019 13:57:39: %FTD-6-302014: Teardown TCP connection 4870 for INSIDE:192.168.0.100/47741 to OUT

```

```
TCP Reset-O from INSIDE
```

```

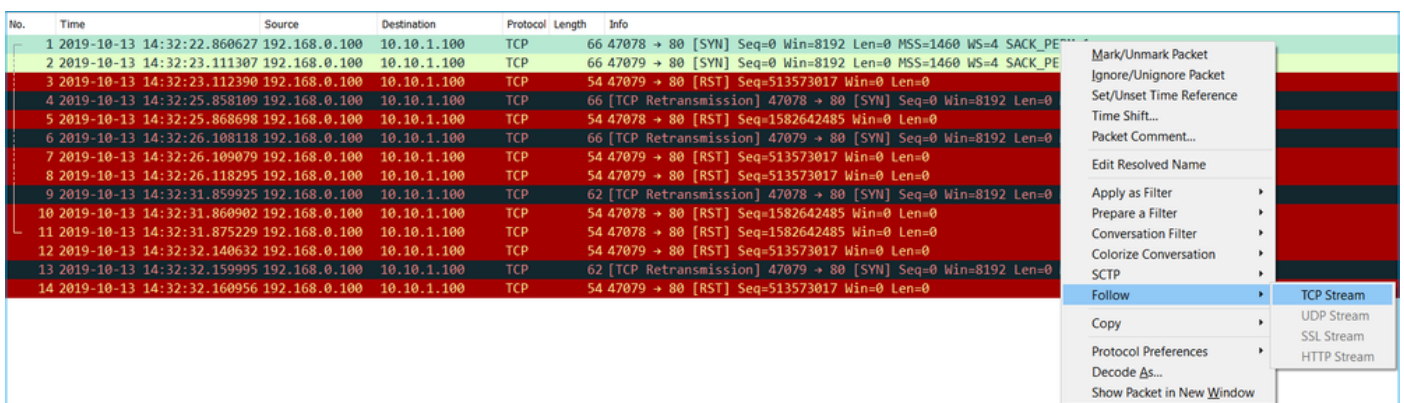
Oct 13 2019 13:57:45: %FTD-6-302013: Built inbound TCP connection 4871 for INSIDE:192.168.0.100/47741 (
Oct 13 2019 13:57:45: %FTD-6-302014: Teardown TCP connection 4871 for INSIDE:192.168.0.100/47741 to OUT

```

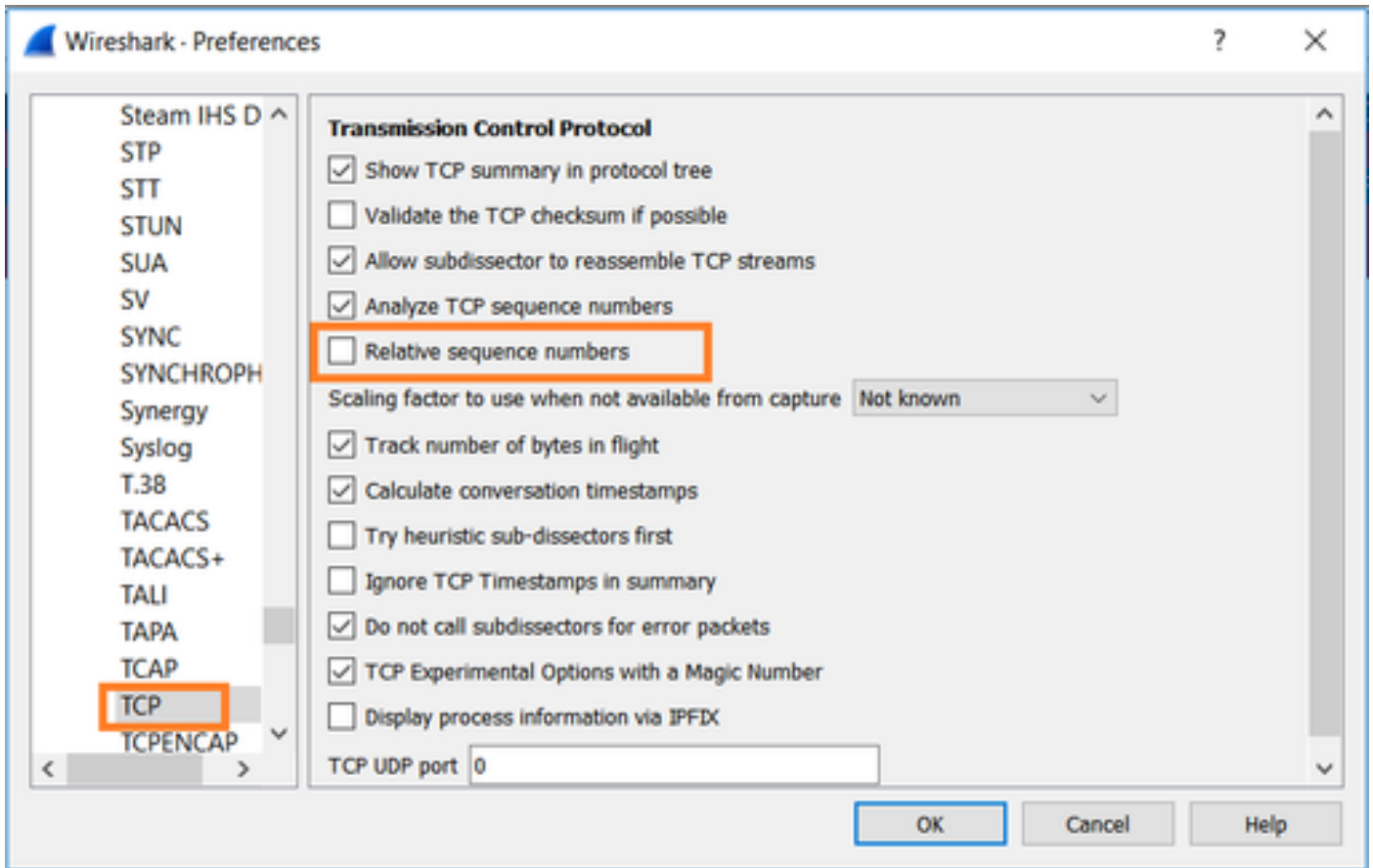
이 로그는 방화벽 INSIDE 인터페이스에 도착하는 TCP RST가 있음을 나타냅니다

Wireshark에서 CAPI 캡처:

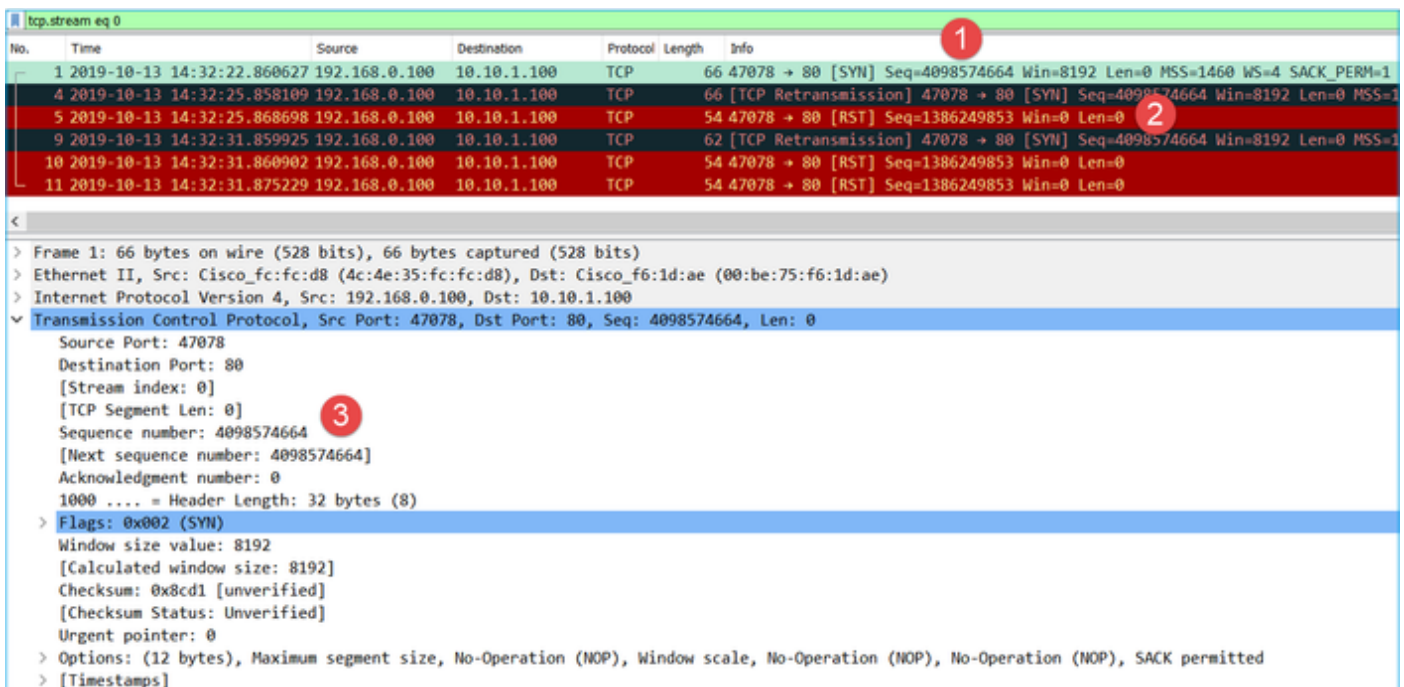
이미지에 표시된 대로 첫 번째 TCP 스트림을 따릅니다.



Wireshark 아래에서 Edit(편집) > Preferences(환경 설정) > Protocols(프로토콜) > TCP로 이동하고 이미지에 표시된 Relative sequence numbers(상대 시퀀스 번호) 옵션의 선택을 취소합니다.



이 그림에서는 CAPI 캡처의 첫 번째 흐름의 내용을 보여 줍니다.



요점:

1. 클라이언트는 TCP SYN 패킷을 전송합니다.
2. 클라이언트는 TCP RST 패킷을 전송합니다.
3. TCP SYN 패킷의 Sequence Number(시퀀스 번호) 값은 4098574664입니다.



CAPO 캡처의 동일한 플로우에는 다음이 포함됩니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-13 14:32:22.860780	192.168.0.100	10.10.1.100	TCP	70	47078 → 80 [SYN] Seq=1386249852 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
4	2019-10-13 14:32:25.858125	192.168.0.100	10.10.1.100	TCP	70	[TCP Retransmission] 47078 → 80 [SYN] Seq=1386249852 Win=8192 Len=0 MSS=1380
5	2019-10-13 14:32:25.868729	192.168.0.100	10.10.1.100	TCP	58	47078 → 80 [RST] Seq=2968892337 Win=0 Len=0

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)  
 > Ethernet II, Src: Cisco\_fc:1d:8e (00:be:75:f6:1d:8e), Dst: Cisco\_fc:fc:d8 (4c:4e:35:fc:fc:d8)  
 > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202  
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100  
 > Transmission Control Protocol, Src Port: 47078, Dst Port: 80, Seq: 1386249852, Len: 0

요점:

1. 클라이언트는 TCP SYN 패킷을 전송합니다. 방화벽은 ISN을 임의로 지정합니다.
2. 클라이언트는 TCP RST 패킷을 전송합니다.

두 가지 캡처를 통해 다음과 같은 결론을 내릴 수 있습니다.

- 클라이언트와 서버 간에는 TCP 3-way 핸드셰이크가 없습니다.
- 클라이언트에서 오는 TCP RST가 있습니다. CAPI 캡처의 TCP RST 시퀀스 번호 값은 1386249853입니다.

## 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. 클라이언트에 대한 캡처를 수행합니다.

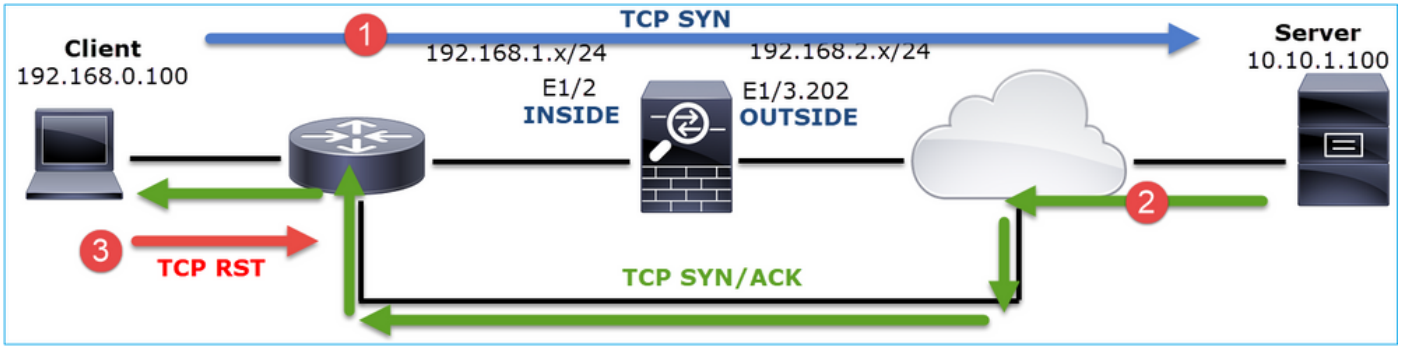
방화벽에서 수집된 캡처를 기반으로 비대칭 플로우에 대한 확실한 지표가 있습니다. 이는 클라이언트가 1386249853(임의 ISN) 값을 사용하여 TCP RST를 전송한다는 사실을 기반으로 합니다.

No.	Time	Source	Destination	Protocol	Length	Info
19	6.040337	192.168.0.100	10.10.1.100	TCP	66	47078→80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
29	9.037499	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 47078→80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1460 WS=4
30	9.048155	10.10.1.100	192.168.0.100	TCP	66	[TCP ACKed unseen segment] 80→47078 [SYN, ACK] Seq=1924342422 Ack=1386249853 Win=0 Len=0
31	9.048184	192.168.0.100	10.10.1.100	TCP	54	47078→80 [RST] Seq=1386249853 Win=0 Len=0

요점:

1. 클라이언트는 TCP SYN 패킷을 전송합니다. 시퀀스 번호는 4098574664이며 방화벽 CAPI(INSIDE interface)에 표시되는 것과 동일합니다
2. ACK 번호가 1386249853(ISN 임의 지정으로 인해 예상됨)인 TCP SYN/ACK가 있습니다. 이 패킷은 방화벽 캡처에서 볼 수 없습니다
3. 클라이언트는 ACK 번호 값이 4098574665인 SYN/ACK가 필요하지만 값이 1386249853인 SYN/ACK를 받았으므로 TCP RST를 보냅니다

이는 다음과 같이 시각화할 수 있습니다.

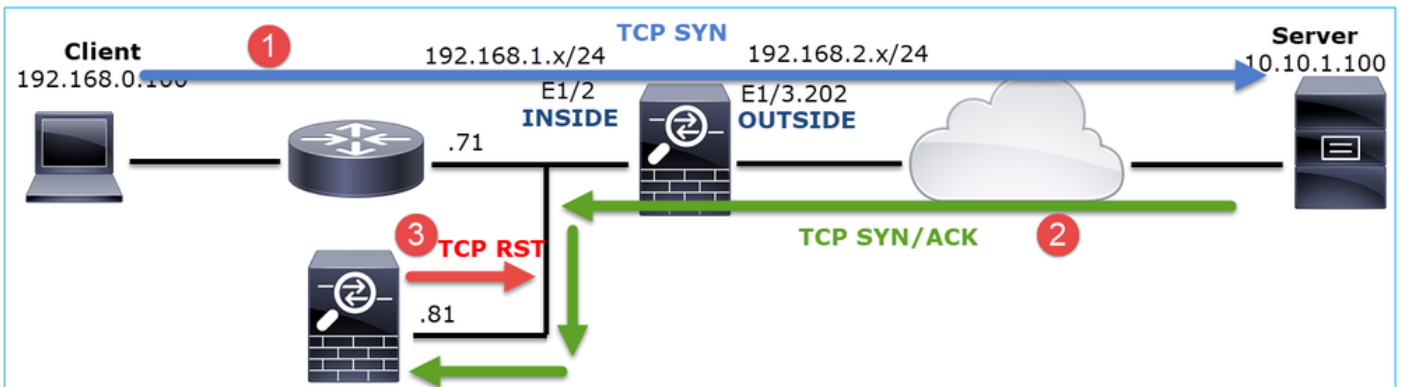


작업 2. 클라이언트와 방화벽 간의 라우팅을 확인합니다.

다음을 확인합니다.

- 캡처에 표시되는 MAC 주소는 예상된 주소입니다.
- 방화벽과 클라이언트 간의 라우팅이 대칭인지 확인합니다.

내부 네트워크에 비대칭 라우팅이 있는 동안 방화벽과 클라이언트 사이에 있는 디바이스에서 RST가 오는 시나리오가 있습니다. 일반적인 경우가 이미지에 표시됩니다.



이 경우 캡처에는 이 내용이 포함됩니다. TCP SYN 패킷의 소스 MAC 주소와 TCP RST의 소스 MAC 주소 및 TCP SYN/ACK 패킷의 목적지 MAC 주소 간의 차이를 확인합니다.

```
<#root>
```

```
firepower#
```

```
show capture CAPI detail
```

```
1: 13:57:36.730217
```

```
4c4e.35fc.fcd8
```

```
00be.75f6.1dae 0x0800 Length: 66
```

```
192.168.0.100.47740 > 10.10.1.100.80: S [tcp sum ok] 3045001876:3045001876(0) win 8192 <mss 1460,
```

```
2: 13:57:36.981104 4c4e.35fc.fcd8 00be.75f6.1dae 0x0800 Length: 66
```

```
192.168.0.100.47741 > 10.10.1.100.80: S [tcp sum ok] 3809380540:3809380540(0) win 8192 <mss 1460,
```

```
3: 13:57:36.981776 00be.75f6.1dae
```

```
a023.9f92.2a4d
```

```
0x0800 Length: 66
```

```
10.10.1.100.80 > 192.168.0.100.47741: S [tcp sum ok] 1304153587:1304153587(0) ack 3809380541 win
```

```
4: 13:57:36.982126
```

a023.9f92.2a4d

00be.75f6.1dae 0x0800 Length: 54  
192.168.0.100.47741 > 10.10.1.100.80:

R

[tcp sum ok] 3809380541:3809380541(0) ack 1304153588 win 8192 (ttl 255, id 48501)  
...

## 사례 5. 느린 TCP 전송(시나리오 1)

문제 설명:

호스트 10.11.4.171과 10.77.19.11 간의 SFTP 전송이 느립니다. 두 호스트 간의 최소 대역폭 (BW)은 100Mbps이지만 전송 속도가 5Mbps를 넘지 않습니다.

이와 동시에 호스트 10.11.2.124와 172.25.18.134 간의 전송 속도는 상당히 더 빠릅니다.

배경 이론:

단일 TCP 플로우의 최대 전송 속도는 BDP(Bandwidth Delay Product)에 의해 결정됩니다. 사용된 공식이 이미지에 표시됩니다.

$$\text{Max Single TCP Flow Throughput [bps]} = \frac{\text{TCP Window (Bytes)}}{\text{RTT (Seconds)}} \times 8 \text{ [bits/Byte]}$$

BDP에 대한 자세한 내용은 다음 리소스를 참조하십시오.

- [링크가 1Gbps인데 애플리케이션이 10Mbps만 사용하는 이유는 무엇입니까?](#)
- [BRKSEC-3021 - 고급 - 방화벽 성능 극대화](#)

시나리오 1. 저속 전송

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: 10.11.4.171

Dst IP: 10.77.19.11

프로토콜: SFTP(FTP over SSH)

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

```
<#root>
```

```
firepower#
```

```
capture CAPI int INSIDE buffer 33554432 match ip host 10.11.4.171 host 10.77.19.11
```

```
firepower#
```

```
capture CAPO int OUTSIDE buffer 33554432 match ip host 10.11.4.171 host 10.77.19.11
```

**⚠ 경고:** FP1xxx 및 FP21xx의 LINA 캡처는 FTD를 통과하는 트래픽의 전송 속도에 영향을 줍니다. 성능(FTD를 통한 전송 속도 저하) 문제를 해결할 때 FP1xxx 및 FP21xxx 플랫폼에서 LINA 캡처를 활성화하지 마십시오. 대신 소스 및 대상 호스트의 캡처와 함께 SPAN 또는 HW Tap

디바이스를 사용합니다. 이 문제는 Cisco 버그 ID CSCvo에 문서화되어 [있습니다30697](#)

```
<#root>
```

```
firepower#
```

```
capture CAPI type raw-data trace interface inside match icmp any any
```

WARNING: Running packet capture can have an adverse impact on performance.

권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.



1	0.000000	10.11.4.171	10.77.19.11	TCP	70	49640 39744 → 22 [SYN] Seq=1737026093 Win=49640 Len=0 MSS=1460 WS=1 SACK_PERM=1
2	0.072521	10.77.19.11	10.11.4.171	TCP	70	49680 22 → 39744 [SYN, ACK] Seq=835172681 Ack=1737026094 Win=49680 Len=0 MSS=1380 WS=1 SACK_PERM=1
3	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680 39744 → 22 [ACK] Seq=1737026094 Ack=835172682 Win=49680 Len=0
4	0.077068	10.77.19.11	10.11.4.171	SSHv2	80	49680 Server: Protocol (SSH-2.0-Sun_SSH_1.1.8)
5	0.000152	10.11.4.171	10.77.19.11	TCP	58	49680 39744 → 22 [ACK] Seq=1737026094 Ack=835172704 Win=49680 Len=0
6	0.000244	10.11.4.171	10.77.19.11	SSHv2	80	49680 Client: Protocol (SSH-2.0-Sun_SSH_1.1.4)
7	0.071545	10.77.19.11	10.11.4.171	TCP	58	49680 22 → 39744 [ACK] Seq=835172704 Ack=1737026116 Win=49680 Len=0
8	0.000153	10.11.4.171	10.77.19.11	SSHv2	538	49680 Client: Key Exchange Init
9	0.041288	10.77.19.11	10.11.4.171	SSHv2	738	49680 Server: Key Exchange Init
10	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680 39744 → 22 [ACK] Seq=1737026596 Ack=835173384 Win=49680 Len=0
11	0.030165	10.77.19.11	10.11.4.171	TCP	58	49680 22 → 39744 [ACK] Seq=835173384 Ack=1737026596 Win=49680 Len=0
12	0.000168	10.11.4.171	10.77.19.11	SSHv2	82	49680 Client: Diffie-Hellman Group Exchange Request

RTT ≈ 80 msec

TCP 창 크기 계산

TCP 패킷을 확장하고, TCP 헤더를 확장하고, Calculated(계산됨) 창 크기를 선택하고, Apply as Column(열로 적용)을 선택합니다.

Transmission Control Protocol, Src Port: 22, Dst Port: 39744, Seq: 835184024, Ack: 1758069308, Len: 32

Source Port: 22  
Destination Port: 39744  
[Stream index: 0]  
[TCP Segment Len: 32]  
Sequence number: 835184024  
[Next sequence number: 835184056]  
Acknowledgment number: 1758069308  
0101 .... = Header Length: 20 bytes (5)  
> Flags: 0x018 (PSH, ACK)  
Window size value: 49680  
[Calculated window size: 49680]  
[Window size scaling factor: ...]  
Checksum: 0x2b49 [unverified]  
[Checksum Status: Unverified]

Expand Subtrees  
Collapse Subtrees  
Expand All  
Collapse All  
Apply as Column

Calculated window size value(계산된 창 크기 값) 열을 확인하여 TCP 세션 중에 최대 창 크기 값이 얼마였는지 확인합니다. 열 이름을 선택하고 값을 정렬할 수도 있습니다.

파일 다운로드를 테스트할 경우(server > client) 서버에서 광고하는 값을 확인해야 합니다. 서버가 광고하는 최대 윈도우 크기 값에 따라 최대 전송 속도가 결정됩니다.

이 경우 TCP 윈도우 크기는 ~50000바이트입니다

No.	Time	Source	Destination	Protocol	Length	Calculated window size	Info
24...	0.000091	10.11.4.171	10.77.19.11	TCP	58	49680	49680 39744 → 22 [ACK] Seq=1758069341 Ack=835173384
24...	0.000077	10.77.19.11	10.11.4.171	TCP	58	49680	49680 22 → 39744 [FIN, ACK] Seq=835184152 Ack=1758069341
24...	0.071605	10.77.19.11	10.11.4.171	TCP	58	49680	49680 22 → 39744 [ACK] Seq=835184152 Ack=1758069341
24...	0.000153	10.11.4.171	10.77.19.11	TCP	58	49680	49680 39744 → 22 [FIN, ACK] Seq=1758069340 Ack=835173384
24...	0.000443	10.11.4.171	10.77.19.11	SSHv2	90		49680 Client: Encrypted packet (len=32)
24...	0.071666	10.77.19.11	10.11.4.171	SSHv2	154		49680 Server: Encrypted packet (len=96)
24...	0.044050	10.11.4.171	10.77.19.11	TCP	58		49680 39744 → 22 [ACK] Seq=1758069308 Ack=835173384
24...	0.073605	10.77.19.11	10.11.4.171	SSHv2	90		49680 Server: Encrypted packet (len=32)
24...	0.000747	10.11.4.171	10.77.19.11	SSHv2	90		49680 Client: Encrypted packet (len=32)

이 값과 대역폭 지연 제품 공식을 사용하여 다음과 같은 조건에서 얻을 수 있는 이론상 최대 대역폭을 구합니다.  $50000 \times 8 / 0.08 = 5\text{Mbps}$  최대 이론상 대역폭.

이는 이 사례에서 고객이 경험하는 것과 일치합니다.

TCP 3-way 핸드셰이크를 자세히 확인합니다. 양쪽 모두, 더 중요하게는 서버는 2^0 = 1(창 크기 조정 없음)을 의미하는 창 크기 값을 0으로 광고합니다. 이는 전송 속도에 부정적인 영향을 미칩니다.

The image shows a Wireshark packet capture of a TCP 3-way handshake. The second packet is a SYN-ACK from the server (10.11.4.171) to the client (10.77.19.11). The packet details show the following information:

- Source Port: 22
- Destination Port: 39744
- Sequence number: 835172681
- Acknowledgment number: 1737026094
- Window size value: 49680
- Flags: 0x012 (SYN, ACK)
- TCP Options:
  - Maximum segment size: 1380 bytes
  - Window scale: 0 (multiply by 1)

이때 서버에서 캡처하고 윈도우 배율을 0으로 광고하는 대상인지 확인하고 다시 구성해야 합니다 (이 방법은 서버 설명서를 참조하십시오).

시나리오 2. 빠른 전송

이제 좋은 시나리오(동일한 네트워크를 통한 빠른 전송)를 살펴보겠습니다.

토폴로지:



관심 흐름:

소스 IP: 10.11.2.124

Dst IP: 172.25.18.134

프로토콜: SFTP(FTP over SSH)

FTD LINA 엔진에서 캡처 활성화

```
<#root>
```

```
firepower#
```

```
capture CAPI int INSIDE buffer 33554432 match ip host 10.11.2.124 host 172.25.18.134
```

firepower#

capture CAPO int OUTSIDE buffer 33554432 match ip host 10.11.2.124 host 172.25.18.134

RTT(Round Trip Time) 계산: 이 경우 RTT는 ≈ 300 msec입니다.

No.	Time	Source	Destination	Protocol	Length
1	0.000000	10.11.2.124	172.25.18.134	TCP	78
2	0.267006	172.25.18.134	10.11.2.124	TCP	78
3	0.000137	10.11.2.124	172.25.18.134	TCP	70
4	0.003784	10.11.2.124	172.25.18.134	SSHv2	91
5	0.266863	172.25.18.134	10.11.2.124	TCP	70
6	0.013580	172.25.18.134	10.11.2.124	SSHv2	91

TCP 창 크기 계산: 서버에서 TCP 창 크기 계수 7을 알립니다.

```
> Internet Protocol Version 4, Src: 172.25.18.134, Dst: 10.11.2.124
Transmission Control Protocol, Src Port: 22, Dst Port: 57093, Seq: 661963571, Ack: 1770516295, Len: 0
  Source Port: 22
  Destination Port: 57093
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 661963571
  [Next sequence number: 661963571]
  Acknowledgment number: 1770516295
  1010 .... = Header Length: 40 bytes (10)
  > Flags: 0x012 (SYN, ACK)
  Window size value: 14480
  [Calculated window size: 14480]
  Checksum: 0x6497 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
    > TCP Option - Maximum segment size: 1300 bytes
    > TCP Option - SACK permitted
    > TCP Option - Timestamps: TSval 390233290, TSecr 981659424
    > TCP Option - No-Operation (NOP)
    > TCP Option - Window scale: 7 (multiply by 128)
  > [SEQ/ACK analysis]
```

서버의 TCP 창 크기는 ≈1600000바이트입니다.

No.	Time	Source	Destination	Protocol	Length	Window size value	Calculated window size	Info
23...	0.002579	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [FIN, ACK]
23...	0.266847	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23...	0.268089	172.25.18.134	10.11.2.124	SSHv2	198	12854	1645312	Server: Encrypted pack
23...	0.000076	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pack
23...	0.000351	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pack
23...	0.000092	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23...	0.000015	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23...	0.000091	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=

Bandwidth Delay Product(대역폭 지연 제품) 공식은 다음 값을 기반으로 합니다.

$$1600000 * 8 / 0.3 = 43\text{Mbps의 이론상 최대 전송 속도}$$



## 사례 6. 느린 TCP 전송(시나리오 2)

문제 설명: 방화벽을 통한 FTP 파일 전송(다운로드)이 느립니다.

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: 192.168.2.220

Dst IP: 192.168.1.220

프로토콜: FTP

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

```
<#root>
```

```
firepower#
```

```
capture CAPI type raw-data buffer 33554432 interface INSIDE match tcp host 192.168.2.220 host 192.168.1.220
```

```
firepower#
```

```
cap CAPO type raw-data buffer 33554432 interface OUTSIDE match tcp host 192.168.2.220 host 192.168.1.220
```

FTP-DATA 패킷을 선택하고 FTD INSIDE CAPTURE(CAPI)의 FTP Data Channel을 따릅니다.

75	0.000412	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670018383
76	0.000518	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
77	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
78	0.000046	192.168.1.220	192.168.2.220	FTP-DATA	[not captured] FTP Data: 124
79	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
80	0.000107	192.168.2.220	192.168.1.220	TCP	Seq=1884231612 Ack=2670019631
81	0.000092	192.168.2.220	192.168.1.220	TCP	Seq=1884231612 Ack=2670020879
82	0.000091	192.168.2.220	192.168.1.220	TCP	
83	0.000015	192.168.2.220	192.168.1.220	TCP	4494 → 2388 [ACK] Seq=188423
84	0.000321	192.168.1.220	192.168.2.220	FTP-DATA	4494 → 2388 [ACK] Seq=188423
85	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
86	0.000153	192.168.2.220	192.168.1.220	TCP	4494 → 2388 [ACK] Seq=188423
87	0.000122	192.168.2.220	192.168.1.220	TCP	4494 → 2388 [ACK] Seq=188423
88	0.918415	192.168.1.220	192.168.2.220	TCP	38 → 54494 [ACK] Seq=2670020
89	0.000397	192.168.2.220	192.168.1.220	TCP	
90	0.000869	192.168.1.220	192.168.2.220	FTP-DATA	(RETR file15mb)

Mark/Unmark Packet

Ignore/Unignore Packet

Set/Unset Time Reference

Time Shift...

Packet Comment...

Edit Resolved Name

Apply as Filter

Prepare a Filter

Conversation Filter

Colorize Conversation

SCTP

Follow

### FTP-DATA 스트림 콘텐츠:


26	0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=357728950 TSecr=0 WS=128
28	1.026564	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
29	1.981594	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2669999678 Ack=1884231612 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
30	0.000488	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669999679 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
34	0.001617	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
35	0.000351	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669999927 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
36	0.000458	192.168.1.220	192.168.2.220	FTP-DATA	[TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
37	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
38	0.000198	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669999927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699993423
39	0.000077	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669999927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699994671
40	0.309906	192.168.1.220	192.168.2.220	TCP	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669999927 Ack=1884231612 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
41	0.000488	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=26699994671 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
42	0.000489	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
43	0.000045	192.168.1.220	192.168.2.220	FTP-DATA	[TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
44	0.000077	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
45	0.000244	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=26699995919 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
46	0.000300	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=26699995919 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=26699997167 SRE=2669999663
47	0.000504	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
48	0.000259	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=26699995919 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=26699997167 SRE=2670000911
49	0.918126	192.168.1.220	192.168.2.220	TCP	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=26699995919 Ack=1884231612 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
50	0.000900	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670000911 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
51	0.000519	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
52	0.000061	192.168.2.220	192.168.1.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	[TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
54	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
55	0.000199	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670002159 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
56	0.000229	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
57	0.000183	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58	0.000106	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=65280 Len=0 TSval=3577292742 TSecr=4264507 SLE=26700004655 SRE=26700007151
59	0.000168	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=68224 Len=0 TSval=3577292743 TSecr=4264507 SLE=26700004655 SRE=26700008399
60	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

CAPO는 다음과 같은 콘텐츠를 캡처합니다.

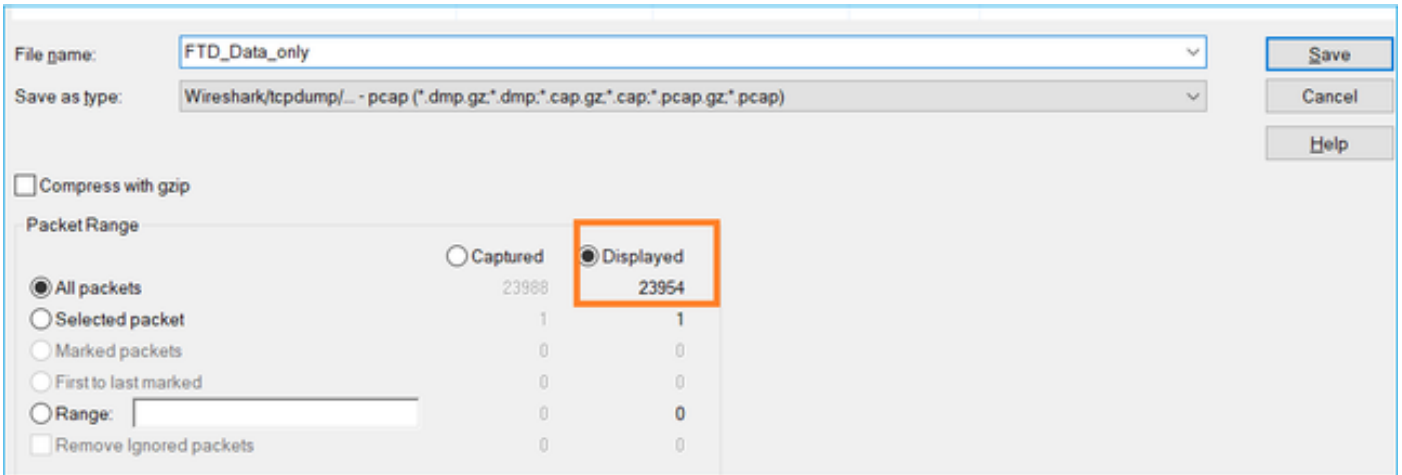
31	0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
33	1.026534	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
34	1.981400	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
35	0.000610	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
38	0.001328	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
40	0.000641	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
41	0.000381	192.168.1.220	192.168.2.220	FTP-DATA	[TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
42	0.000046	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
43	0.000290	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=2224319408 SRE=2224320656
44	0.000076	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=2224321904
45	0.309905	192.168.1.220	192.168.2.220	TCP	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224323152 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
46	0.000590	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
47	0.000412	192.168.1.220	192.168.2.220	FTP-DATA	[TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
49	0.000076	192.168.2.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
50	0.000290	192.168.1.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
51	0.000046	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=2224324400 SRE=2224326896
52	0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000351	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2224324400 SRE=2224328144
54	0.918019	192.168.1.220	192.168.2.220	TCP	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224323152 Ack=2157030682 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
55	0.001007	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224328144 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
56	0.000457	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
57	0.000061	192.168.2.220	192.168.1.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58	0.000016	192.168.1.220	192.168.2.220	FTP-DATA	[TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
59	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
60	0.000274	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224329392 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
61	0.000214	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224330640 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
62	0.000122	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
63	0.000168	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224330640 Win=65280 Len=0 TSval=3577292742 TSecr=4264507 SLE=2224331888 SRE=2224334384
64	0.000107	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

요점:

1. TCP OOO(Out-Of-Order) 패킷이 있습니다.
2. TCP 재전송이 있습니다.
3. 패킷 손실(삭제된 패킷)을 나타냅니다.

 팁: File(파일) > Export Specified Packets(지정된 패킷 내보내기)로 이동할 때 캡처를 저장함

니다. 그런 다음 표시된 패킷 범위만 저장합니다



## 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

### 작업 1. 패킷 손실 위치를 식별합니다.

이와 같은 경우 동시 캡처를 수행하고 Divide and Conquer 방법론을 사용하여 패킷 손실의 원인이 되는 네트워크 세그먼트를 식별해야 합니다. 방화벽 관점에서 보면 다음과 같은 3가지 주요 시나리오가 있습니다.

1. 패킷 손실은 방화벽 자체로 인해 발생합니다.
2. 패킷 손실은 방화벽 디바이스(서버에서 클라이언트로 가는 방향)의 다운스트림으로 발생합니다.
3. 패킷 손실은 방화벽 디바이스 업스트림에서 발생합니다(클라이언트에서 서버로 방향).

방화벽으로 인한 패킷 손실: 패킷 손실이 방화벽으로 인한 것인지 확인하려면 인그레스 캡처를 이그레스 캡처와 비교해야 합니다. 두 개의 서로 다른 캡처를 비교하는 방법은 꽤 많다. 이 단원에서는 이 작업을 수행하는 한 가지 방법을 보여 줍니다.

### 패킷 손실을 식별하기 위해 2개의 캡처를 비교하는 절차

1단계. 2 캡처에 동일한 타임 윈도우의 패킷이 포함되어 있는지 확인합니다. 즉, 한 캡처에는 다른 캡처 전이나 후에 캡처된 패킷이 없어야 합니다. 다음과 같은 몇 가지 방법이 있습니다.

- 첫 번째와 마지막 패킷 IP ID(IP ID) 값을 확인합니다.
- 첫 번째 및 마지막 패킷 타임스탬프 값을 확인합니다.

이 예에서는 각 캡처의 첫 번째 패킷이 동일한 IP ID 값을 갖는다는 것을 확인할 수 있습니다.

동일하지 않은 경우 다음을 수행합니다.

1. 각 캡처의 첫 번째 패킷에서 타임스탬프를 비교합니다.
2. 최신 Timestamp가 있는 캡처에서 필터를 가져오면 Timestamp 필터가 ==부터 >=(첫 번째 패킷) 및 <=(마지막 패킷)로 변경됩니다. 예:

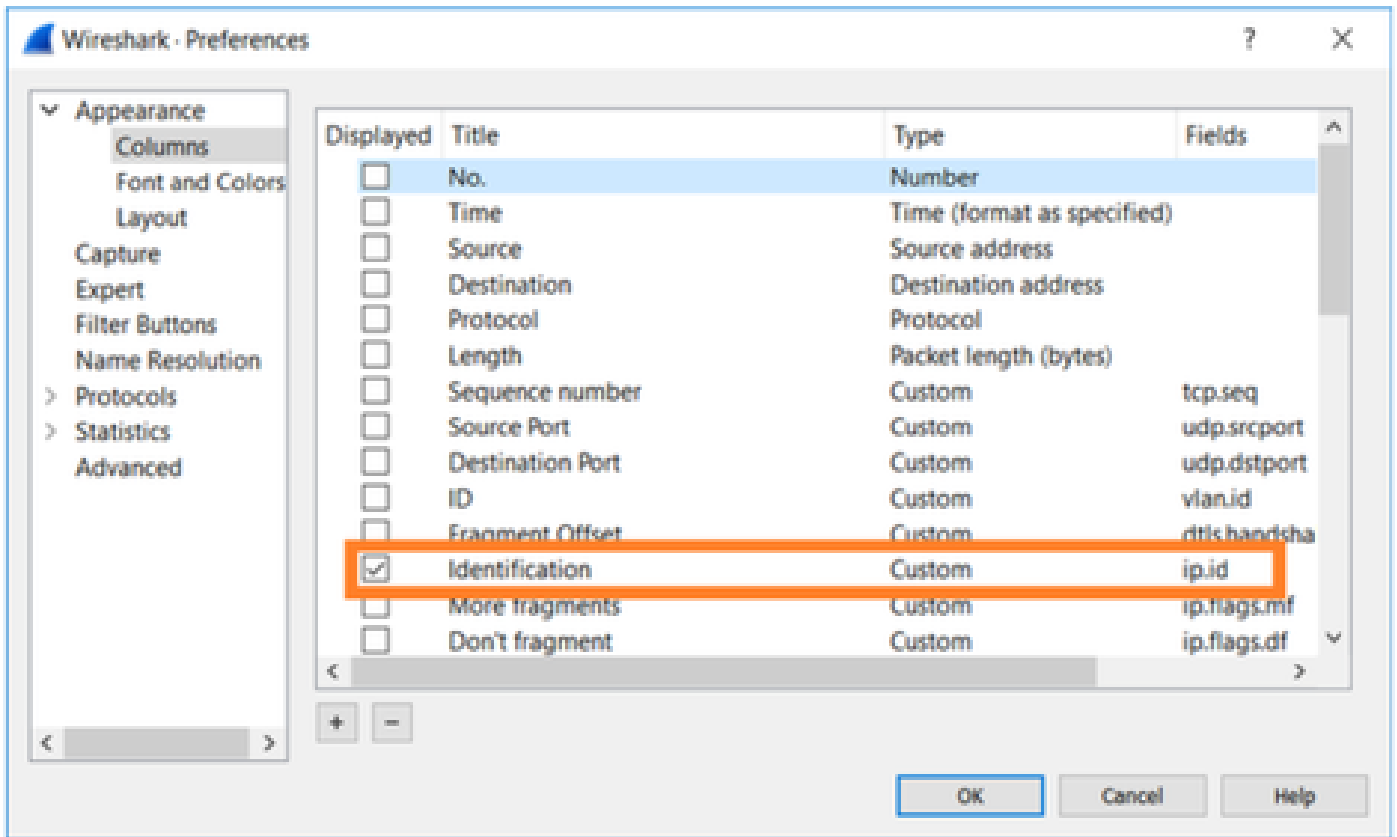
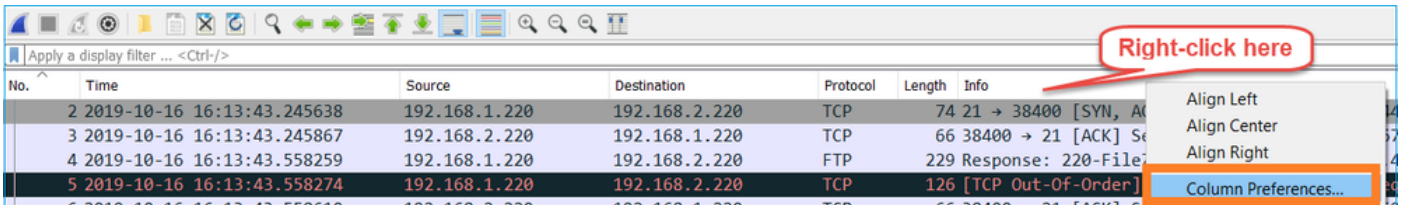
(frame.time >= "2019년 10월 16일 16:13:43.244692000") &&(frame.time <= "2019년 10월 16일 16:20:21.785130000")

3. 지정된 패킷을 새 캡처로 내보내고, File > Export Specified Packets를 선택한 다음 Displayed 패킷을 저장합니다. 이때 두 캡처에는 동일한 타임 윈도우를 커버하는 패킷이 포함되어야 합니다. 이제 2개의 캡처를 비교할 수 있습니다.

2단계. 두 캡처 간의 비교에 사용할 패킷 필드를 지정합니다. 사용할 수 있는 필드의 예:

- IP 식별
- RTP 시퀀스 번호
- ICMP 시퀀스 번호

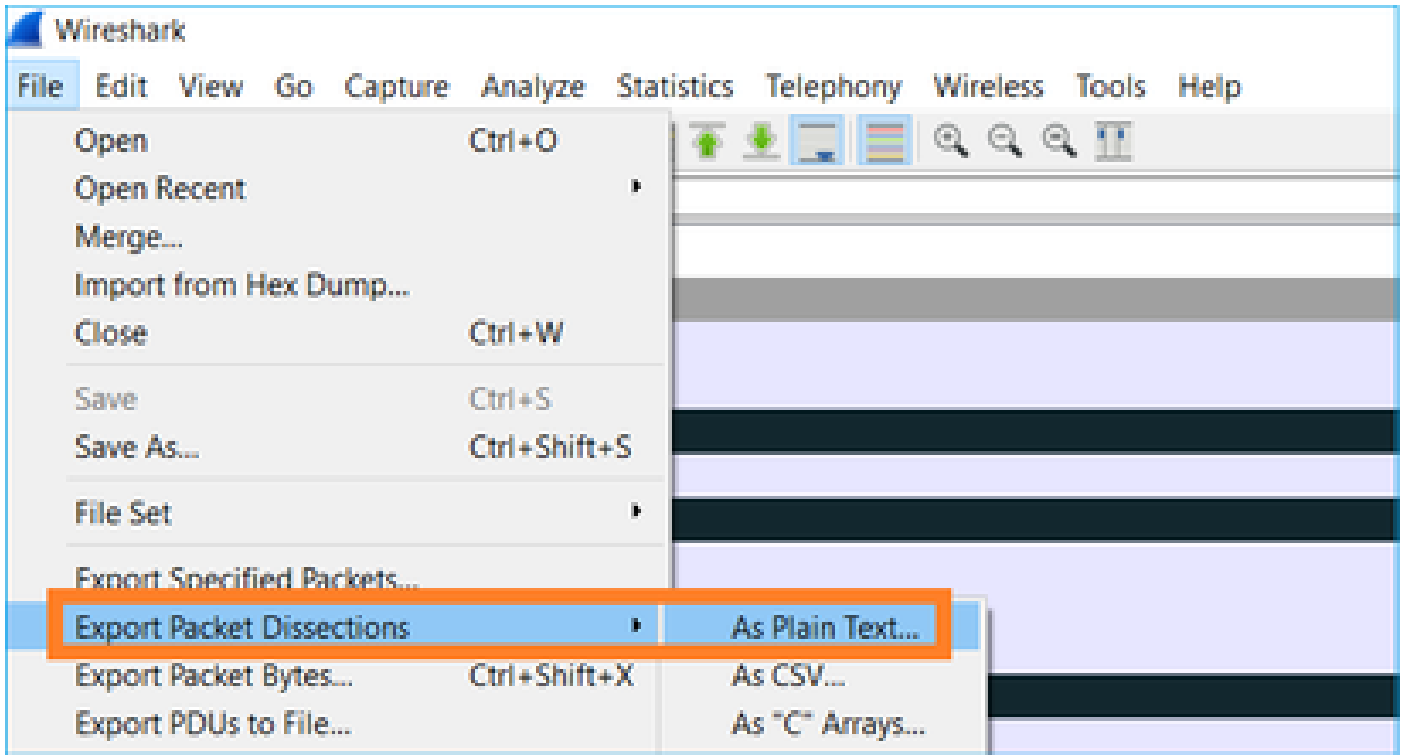
1단계에서 지정한 각 패킷의 필드가 포함된 각 캡처의 텍스트 버전을 생성합니다. 이렇게 하려면 관심 있는 열만 남겨 둡니다. 예를 들어, IP ID를 기준으로 패킷을 비교하려는 경우 이미지에 표시된 대로 캡처를 수정합니다.



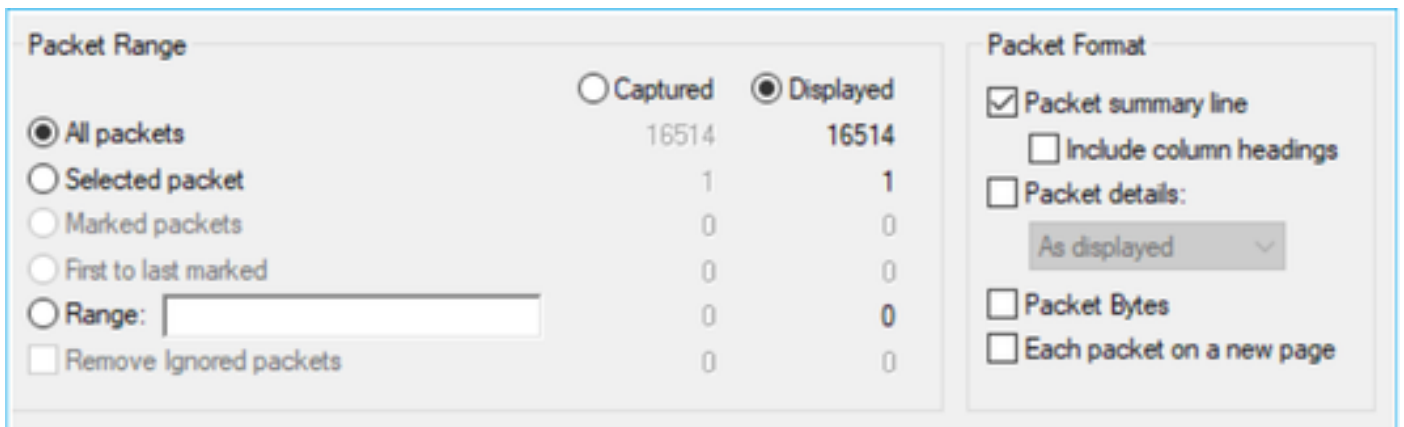
결과:

Identification
0x150e (5398)
0xfdb0 (64944)
0x1512 (5394)
<b>0x1510 (5392)</b>
0xfdb1 (64945)
<b>0xfdb2 (64946)</b>
0xfdb3 (64947)
0x1513 (5395)
0xfdb4 (64948)
<b>0xfdb5 (64949)</b>
0x1516 (5398)
<b>0x1515 (5397)</b>
0xfdb6 (64950)
0x1517 (5399)
0xfdb7 (64951)
0x1518 (5400)
0xfdb8 (64952)
<b>0xfdb9 (64953)</b>
0x151b (5403)
<b>0x151a (5402)</b>
0xfdba (64954)
0x151c (5404)
0xfdbb (64955)
0x151d (5405)
0x0a34 (2612)
0xfdbc (64956)
<b>0x0a35 (2613)</b>
0x151f (5407)
0x0a36 (2614)
<ul style="list-style-type: none"> <li>Frame 23988: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) <ul style="list-style-type: none"> <li>Encapsulation type: Ethernet (1) <ul style="list-style-type: none"> <li>Arrival Time: Oct 16, 2019 16:20:21.785130000 Central European Daylight Time</li> </ul> </li> </ul> </li> </ul>

3단계. 이미지에 표시된 대로 캡처의 텍스트 버전(File > Export Packet Dissections > As Plain Text...)을 생성합니다.



이미지에 표시된 것처럼 표시된 필드의 값만 내보내려면 Include column headings and Packet details(열 머리글 및 패킷 세부사항 포함) 옵션의 선택을 취소합니다.

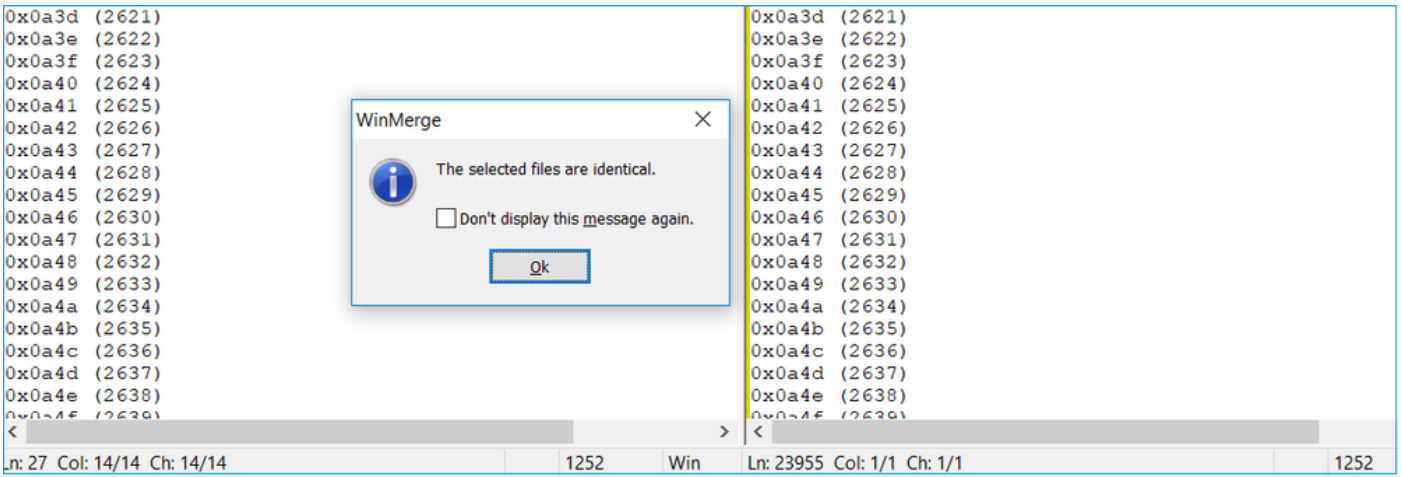


4단계. 파일의 패킷을 정렬합니다. Linux sort 명령을 사용하여 다음을 수행할 수 있습니다.

```
<#root>
#
sort CAPI_IDs > file1.sorted
#
sort CAPO_IDs > file2.sorted
```

5단계. 텍스트 비교 도구(예: WinMerge) 또는 Linux diff 명령을 사용하여 두 캡처 간의 차이를 확인

합니다.



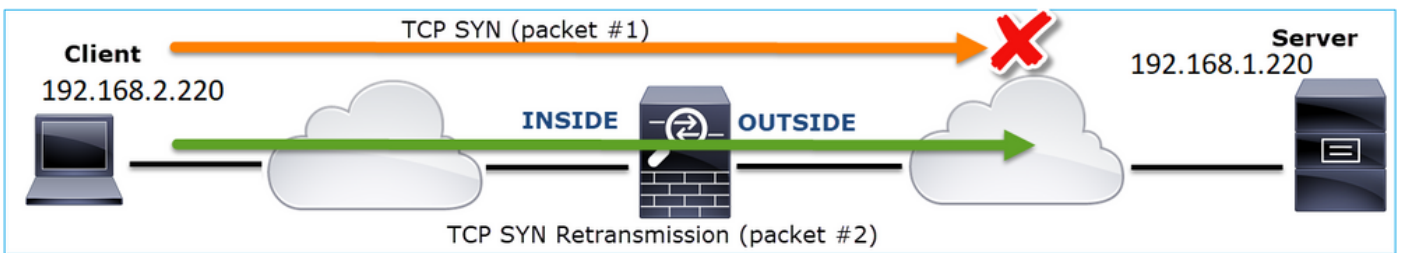
이 경우 FTP 데이터 트래픽에 대한 CAPI 및 CAPO 캡처는 동일합니다. 이는 패킷 손실이 방화벽에 의해 발생하지 않았음을 입증합니다.

업스트림/다운스트림 패킷 손실을 식별합니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-16 16:13:44.169516	192.168.2.220	192.168.1.220	TCP	74	54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=1
2	2019-10-16 16:13:45.196050	192.168.2.220	192.168.1.220	TCP	74	[TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=1
3	2019-10-16 16:13:47.177450	192.168.1.220	192.168.2.220	TCP	74	2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=3577291508 TSecr=3577291508
4	2019-10-16 16:13:47.178060	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
5	2019-10-16 16:13:47.179388	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224316912 Ack=2157030682 Win=66048 Len=1248 TSval=4264384 TSecr=3577291508
6	2019-10-16 16:13:47.180029	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
7	2019-10-16 16:13:47.180410	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224319408 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291510
8	2019-10-16 16:13:47.180456	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224320656 Ack=2157030682 Win=66048 Len=1248 TSval=4264384 TSecr=3577291510
9	2019-10-16 16:13:47.180746	192.168.2.220	192.168.1.220	TCP	78	[TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=35072 Len=0 TSval=3577291510 TSecr=4264415
10	2019-10-16 16:13:47.180822	192.168.2.220	192.168.1.220	TCP	78	[TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=37888 Len=0 TSval=3577291510 TSecr=4264415
11	2019-10-16 16:13:47.489827	192.168.1.220	192.168.2.220	TCP	1314	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224318160 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291510
12	2019-10-16 16:13:47.490407	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
13	2019-10-16 16:13:47.490819	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224321904 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
14	2019-10-16 16:13:47.490880	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224324400 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
15	2019-10-16 16:13:47.490956	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224325648 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
16	2019-10-16 16:13:47.491246	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415

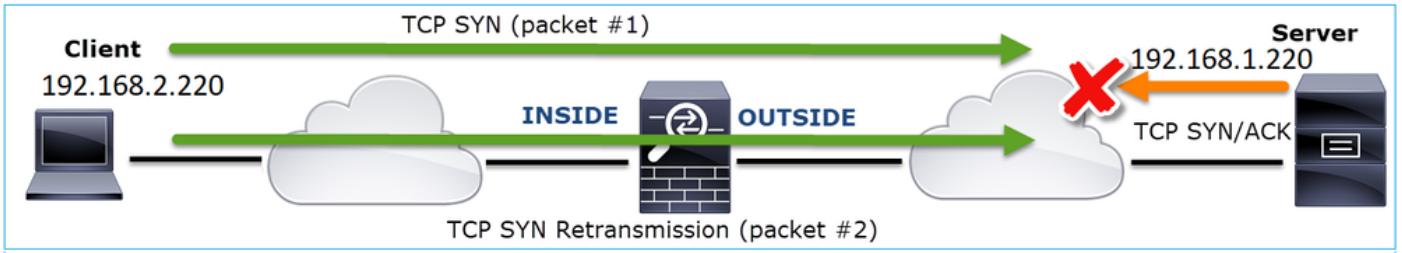
요점:

1. 이 패킷은 TCP 재전송입니다. 특히 패시브 모드의 FTP 데이터에 대해 클라이언트에서 서버로 전송되는 TCP SYN 패킷입니다. 클라이언트가 패킷을 재전송하고 초기 SYN(패킷 #1)을 볼 수 있으므로 패킷이 방화벽으로 업스트림에서 손실되었습니다.

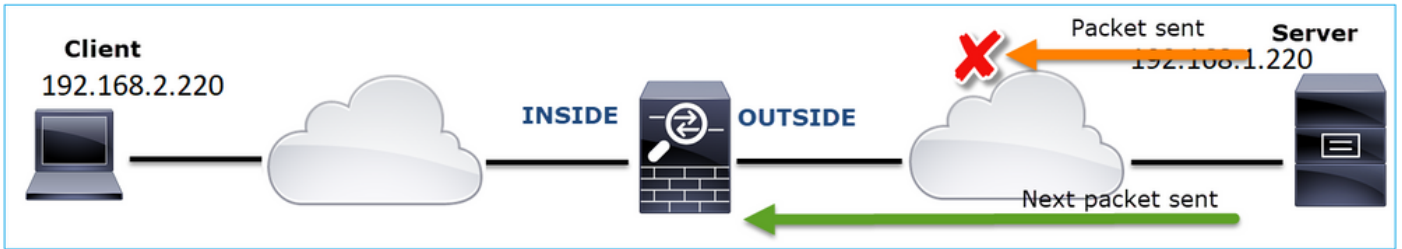


이 경우 SYN 패킷이 서버에 도착했지만 돌아오는 도중에 SYN/ACK 패킷이 손실되었을 수 있습니다.





2. 이전 세그먼트가 확인/캡처되지 않은 것으로 확인된 서버 및 Wireshark의 패킷이 있습니다. 캡처되지 않은 패킷이 서버에서 클라이언트로 전송되었으며 방화벽 캡처에서 보이지 않기 때문에 서버와 방화벽 간에 패킷이 손실되었습니다.



이는 FTP 서버와 방화벽 간에 패킷 손실이 있음을 나타냅니다.

## 작업 2. 추가 캡처

엔드포인트에서 캡처와 함께 추가 캡처를 생성합니다. 패킷 손실의 원인이 되는 문제가 있는 세그먼트를 더 격리하려면 Divide and Conquer 방법을 적용해 보십시오.

No.	Time	Source	Destination	Protocol	Length	Info
155	2019-10-16 16:13:51.749845	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
156	2019-10-16 16:13:51.749860	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
157	2019-10-16 16:13:51.749872	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
158	2019-10-16 16:13:51.750722	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224385552 Win=180480 Len=0 TSv
159	2019-10-16 16:13:51.750744	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
160	2019-10-16 16:13:51.750768	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800 Win=183424 Len=0 TSv
161	2019-10-16 16:13:51.750782	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
162	2019-10-16 16:13:51.751001	192.168.2.220	192.168.1.220	TCP	70	[TCP Dup ACK 160#1] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
163	2019-10-16 16:13:51.751024	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
164	2019-10-16 16:13:51.751378	192.168.2.220	192.168.1.220	TCP	70	[TCP Dup ACK 160#2] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
165	2019-10-16 16:13:51.751402	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
166	2019-10-16 16:13:51.751622	192.168.2.220	192.168.1.220	TCP	70	[TCP Dup ACK 160#3] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
167	2019-10-16 16:13:51.751648	192.168.1.220	192.168.2.220	FTP-DA..	1314	[TCP Fast Retransmission] FTP Data: 1248 bytes (PASV) (RETR file15mb)

> Frame 167: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits) on interface 0  
 > Ethernet II, Src: Vmware\_30:2b:78 (00:0c:29:30:2b:78), Dst: Cisco\_9d:89:9b (50:3d:e5:9d:89:9b)  
 > Internet Protocol Version 4, Src: 192.168.1.220, Dst: 192.168.2.220  
 > Transmission Control Protocol, Src Port: 2388, Dst Port: 494, Seq: 2224386800, Ack: 2157030682, Len: 1248  
 FTP Data (1248 bytes data)  
 [Setup frame: 33]  
 [Setup method: PASV]  
 [Command: RETR file15mb]  
 Command frame: 40  
 [Current working directory: /]  
 > Line-based text data (1 lines)

## 요점:

1. 수신기(이 경우 FTP 클라이언트)는 들어오는 TCP 시퀀스 번호를 추적합니다. 패킷이 누락되었음을 감지하면(예상 시퀀스 번호를 건너뛰었음) ACK='건너뛴 예상 시퀀스 번호'로 ACK 패킷을 생성합니다. 이 예에서는 Ack=2224386800입니다.
2. Dup ACK는 TCP 빠른 재전송(Duplicate ACK 수신 후 20msec 이내 재전송)을 트리거합니다.

중복 ACK는 무엇을 의미합니까?

- 몇 개의 중복 ACK가 있지만 실제 재전송이 없을 경우 무질서하게 도착하는 패킷이 있을 가능성이 높습니다.
- 실제 재전송이 뒤따르는 중복 ACK는 패킷 손실이 어느 정도 있음을 나타냅니다.

작업 3. 전송 패킷에 대한 방화벽 처리 시간을 계산합니다.

서로 다른 두 인터페이스에 동일한 캡처를 적용합니다.

```
<#root>
```

```
firepower#
```

```
capture CAPI buffer 33554432 interface INSIDE match tcp host 192.168.2.220 host 192.168.1.220
```

```
firepower#
```

```
capture CAPI interface OUTSIDE
```

캡처 내보내기: 인그레스 패킷과 이그레스 패킷의 시간 차이 확인

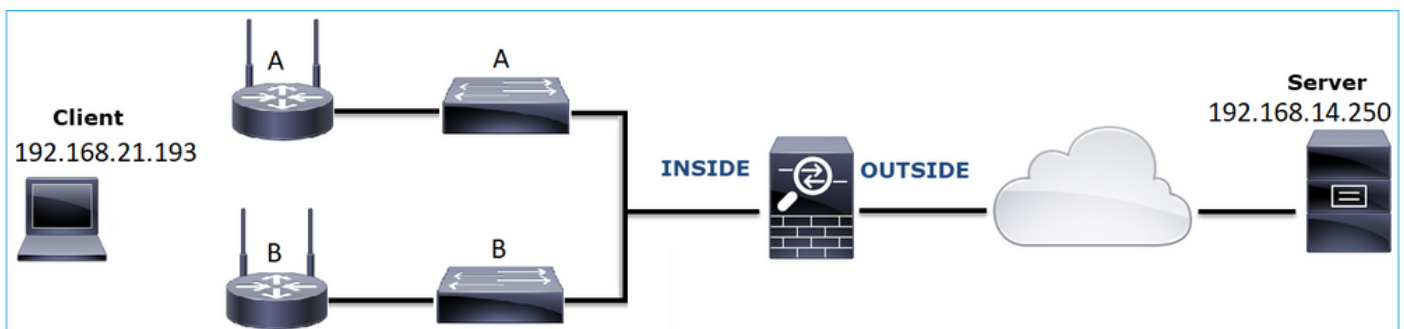
## 사례 7. TCP 연결 문제(패킷 손상)

문제 설명:

무선 클라이언트(192.168.21.193)가 대상 서버(192.168.14.250 - HTTP)에 연결하려고 시도하며 두 가지 시나리오가 있습니다.

- 클라이언트가 액세스 포인트(AP) 'A'에 연결되면 HTTP 연결이 작동하지 않습니다.
- 클라이언트가 액세스 포인트(AP) 'B'에 연결되면 HTTP 연결이 작동합니다.

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: 192.168.21.193

Dst IP: 192.168.14.250

프로토콜: TCP 80

### 캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

<#root>

firepower#

```
capture CAPI int INSIDE match ip host 192.168.21.193 host 192.168.14.250
```

firepower#

```
capture CAPO int OUTSIDE match ip host 192.168.21.193 host 192.168.14.250
```

### 캡처 - 기능 시나리오:

베이스라인으로서 정상 작동이 확인된 시나리오의 캡처를 사용하는 것은 항상 매우 유용합니다.

이 그림에서는 NGFW INSIDE 인터페이스에서 캡처한 내용을 보여줍니다

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 17:03:25.554582	192.168.21.193	192.168.14.250	TCP	66	1055 → 80 [SYN] Seq=1341231 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 17:03:25.555238	192.168.14.250	192.168.21.193	TCP	66	80 → 1055 [SYN, ACK] Seq=1015787006 Ack=1341232 Win=64240 Len=0 MSS=1380 SACK_PERM=1
3	2013-08-08 17:03:25.579910	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1341232 Ack=1015787007 Win=65535 Len=0
4	2013-08-08 17:03:25.841081	192.168.21.193	192.168.14.250	HTTP	370	GET /ttest.html HTTP/1.1
5	2013-08-08 17:03:25.848466	192.168.14.250	192.168.21.193	TCP	1438	80 → 1055 [ACK] Seq=1015787007 Ack=1341544 Win=63928 Len=1380 [TCP segment of a reassembled PDU]
6	2013-08-08 17:03:25.848527	192.168.14.250	192.168.21.193	HTTP	698	HTTP/1.1 404 Not Found (text/html)
7	2013-08-08 17:03:25.858445	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1341544 Ack=1015789027 Win=65535 Len=0
8	2013-08-08 17:03:34.391749	192.168.21.193	192.168.14.250	HTTP	369	GET /test.html HTTP/1.1
9	2013-08-08 17:03:34.395487	192.168.14.250	192.168.21.193	HTTP	586	HTTP/1.1 200 OK (text/html)
10	2013-08-08 17:03:34.606352	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1341855 Ack=1015789555 Win=65007 Len=0
11	2013-08-08 17:03:40.739601	192.168.21.193	192.168.14.250	HTTP	483	GET /test.html HTTP/1.1
12	2013-08-08 17:03:40.741538	192.168.14.250	192.168.21.193	HTTP	271	HTTP/1.1 304 Not Modified

이 그림에서는 NGFW OUTSIDE 인터페이스에서 캡처한 내용을 보여줍니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 17:03:25.554872	192.168.21.193	192.168.14.250	TCP	66	1055 → 80 [SYN] Seq=1839800324 Win=65535 Len=0 MSS=1380 SACK_PERM=1
2	2013-08-08 17:03:25.555177	192.168.14.250	192.168.21.193	TCP	66	80 → 1055 [SYN, ACK] Seq=521188628 Ack=1839800325 Win=64240 Len=0 MSS=1460 SACK_PERM=1
3	2013-08-08 17:03:25.579926	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1839800325 Ack=521188629 Win=65535 Len=0
4	2013-08-08 17:03:25.841112	192.168.21.193	192.168.14.250	HTTP	370	GET /ttest.html HTTP/1.1
5	2013-08-08 17:03:25.848451	192.168.14.250	192.168.21.193	TCP	1438	80 → 1055 [ACK] Seq=521188629 Ack=1839800637 Win=63928 Len=1380 [TCP segment of a reassembled PDU]
6	2013-08-08 17:03:25.848512	192.168.14.250	192.168.21.193	HTTP	698	HTTP/1.1 404 Not Found (text/html)
7	2013-08-08 17:03:25.858476	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1839800637 Ack=521190649 Win=65535 Len=0
8	2013-08-08 17:03:34.391779	192.168.21.193	192.168.14.250	HTTP	369	GET /test.html HTTP/1.1
9	2013-08-08 17:03:34.395456	192.168.14.250	192.168.21.193	HTTP	586	HTTP/1.1 200 OK (text/html)
10	2013-08-08 17:03:34.606368	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1839800948 Ack=521191177 Win=65007 Len=0
11	2013-08-08 17:03:40.739646	192.168.21.193	192.168.14.250	HTTP	483	GET /test.html HTTP/1.1
12	2013-08-08 17:03:40.741523	192.168.14.250	192.168.21.193	HTTP	271	HTTP/1.1 304 Not Modified

요점:

1. 2개의 캡처는 거의 동일합니다(ISN 임의 지정 고려).
2. 패킷 손실의 징후가 없습니다.
3. OOO(No Out-Of-Order) 패킷
4. 3개의 HTTP GET 요청이 있습니다. 첫 번째는 404 'Not Found', 두 번째는 200 'OK', 세 번째는 304 'Not Modified' 리디렉션 메시지를 받습니다.

캡처 - 알려진 결함 시나리오:

인그레스 캡처(CAPI) 내용입니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909193	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 15:33:31.909849	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=867575959 Ack=4231766829 Win=64240 Len=0 MSS=1380 SACK_PERM=1
3	2013-08-08 15:33:31.913267	192.168.21.193	192.168.14.250	TCP	60	3072 → 80 [ACK] Seq=4231766829 Ack=867575960 Win=65535 Len=2[Malformed Packet]
4	2013-08-08 15:33:31.913649	192.168.14.250	192.168.21.193	HTTP	222	HTTP/1.1 400 Bad Request (text/html)
5	2013-08-08 15:33:31.980326	192.168.21.193	192.168.14.250	TCP	369	[TCP Retransmission] 3072 → 80 [PSH, ACK] Seq=4231766829 Ack=867575960 Win=65535 Len=311
6	2013-08-08 15:33:32.155723	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3072 [ACK] Seq=867576125 Ack=4231767140 Win=63929 Len=0
7	2013-08-08 15:33:34.871460	192.168.14.250	192.168.21.193	TCP	222	[TCP Retransmission] 80 → 3072 [FIN, PSH, ACK] Seq=867575960 Ack=4231767140 Win=63929 Len=164
8	2013-08-08 15:33:34.894713	192.168.21.193	192.168.14.250	TCP	60	3072 → 80 [ACK] Seq=4231767140 Ack=867576125 Win=65371 Len=2
9	2013-08-08 15:33:34.933560	192.168.21.193	192.168.14.250	TCP	60	[TCP Retransmission] 3072 → 80 [FIN, ACK] Seq=4231767140 Ack=867576125 Win=65371 Len=2
10	2013-08-08 15:33:34.933789	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3072 [ACK] Seq=867576125 Ack=4231767143 Win=63927 Len=0
11	2013-08-08 15:33:35.118234	192.168.21.193	192.168.14.250	TCP	66	3073 → 80 [SYN] Seq=2130836820 Win=65535 Len=0 MSS=1460 SACK_PERM=1
12	2013-08-08 15:33:35.118737	192.168.14.250	192.168.21.193	TCP	66	80 → 3073 [SYN, ACK] Seq=2991287216 Ack=2130836821 Win=64240 Len=0 MSS=1380 SACK_PERM=1
13	2013-08-08 15:33:35.121575	192.168.21.193	192.168.14.250	TCP	60	3073 → 80 [ACK] Seq=2130836821 Ack=2991287217 Win=65535 Len=2[Malformed Packet]
14	2013-08-08 15:33:35.121621	192.168.21.193	192.168.14.250	TCP	371	[TCP Out-Of-Order] 3073 → 80 [PSH, ACK] Seq=2130836821 Ack=2991287217 Win=65535 Len=313
15	2013-08-08 15:33:35.121896	192.168.14.250	192.168.21.193	HTTP	222	HTTP/1.1 400 Bad Request (text/html)
16	2013-08-08 15:33:35.124657	192.168.21.193	192.168.14.250	TCP	60	3073 → 80 [ACK] Seq=2130837134 Ack=2991287382 Win=65371 Len=2
17	2013-08-08 15:33:35.124840	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3073 [ACK] Seq=2991287382 Ack=2130837136 Win=63925 Len=0
18	2013-08-08 15:33:35.126046	192.168.21.193	192.168.14.250	TCP	60	[TCP Spurious Retransmission] 3073 → 80 [FIN, ACK] Seq=2130837134 Ack=2991287382 Win=65371 Len=2
19	2013-08-08 15:33:35.126244	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3073 [ACK] Seq=2991287382 Ack=2130837137 Win=63925 Len=0

요점:

1. TCP 3-way 핸드셰이크가 있습니다.
2. TCP 재전송과 패킷 손실의 징후가 있습니다.
3. Wireshark에 의해 형식이 잘못된 것으로 식별되는 패킷(TCP ACK)이 있습니다.

이 그림에서는 이그레스 캡처(CAPO) 내용을 보여 줍니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909514	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=230342488 Win=65535 Len=0 MSS=1380 SACK_PERM=1
2	2013-08-08 15:33:31.909804	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=268013986 Ack=230342489 Win=64240 Len=0 MSS=1460 SACK_PERM=1
3	2013-08-08 15:33:31.913298	192.168.21.193	192.168.14.250	TCP	60	3072 → 80 [ACK] Seq=230342489 Ack=268013987 Win=65535 Len=2[Malformed Packet]
4	2013-08-08 15:33:31.913633	192.168.14.250	192.168.21.193	HTTP	222	HTTP/1.1 400 Bad Request (text/html)
5	2013-08-08 15:33:31.980357	192.168.21.193	192.168.14.250	TCP	369	[TCP Retransmission] 3072 → 80 [PSH, ACK] Seq=230342489 Ack=268013987 Win=65535 Len=311
6	2013-08-08 15:33:32.155692	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3072 [ACK] Seq=268014152 Ack=230342800 Win=63929 Len=0
7	2013-08-08 15:33:34.871430	192.168.14.250	192.168.21.193	TCP	222	[TCP Retransmission] 80 → 3072 [FIN, PSH, ACK] Seq=268013987 Ack=230342800 Win=63929 Len=164
8	2013-08-08 15:33:34.894759	192.168.21.193	192.168.14.250	TCP	60	3072 → 80 [ACK] Seq=230342800 Ack=268014152 Win=65371 Len=2
9	2013-08-08 15:33:34.933575	192.168.21.193	192.168.14.250	TCP	60	[TCP Retransmission] 3072 → 80 [FIN, ACK] Seq=230342800 Ack=268014152 Win=65371 Len=2
10	2013-08-08 15:33:34.933774	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3072 [ACK] Seq=268014152 Ack=230342803 Win=63927 Len=0
11	2013-08-08 15:33:35.118524	192.168.21.193	192.168.14.250	TCP	66	3073 → 80 [SYN] Seq=2731219422 Win=65535 Len=0 MSS=1380 SACK_PERM=1
12	2013-08-08 15:33:35.118707	192.168.14.250	192.168.21.193	TCP	66	80 → 3073 [SYN, ACK] Seq=2453407925 Ack=2731219423 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13	2013-08-08 15:33:35.121591	192.168.21.193	192.168.14.250	TCP	60	3073 → 80 [ACK] Seq=2731219423 Ack=2453407926 Win=65535 Len=2[Malformed Packet]
14	2013-08-08 15:33:35.121652	192.168.21.193	192.168.14.250	TCP	371	[TCP Out-Of-Order] 3073 → 80 [PSH, ACK] Seq=2731219423 Ack=2453407926 Win=65535 Len=313
15	2013-08-08 15:33:35.121865	192.168.14.250	192.168.21.193	HTTP	222	HTTP/1.1 400 Bad Request (text/html)
16	2013-08-08 15:33:35.124673	192.168.21.193	192.168.14.250	TCP	60	3073 → 80 [ACK] Seq=2731219736 Ack=2453408091 Win=65371 Len=2
17	2013-08-08 15:33:35.124810	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3073 [ACK] Seq=2453408091 Ack=2731219738 Win=63925 Len=0
18	2013-08-08 15:33:35.126061	192.168.21.193	192.168.14.250	TCP	60	[TCP Spurious Retransmission] 3073 → 80 [FIN, ACK] Seq=2731219736 Ack=2453408091 Win=65371 Len=2
19	2013-08-08 15:33:35.126229	192.168.14.250	192.168.21.193	TCP	58	[TCP ACKed unseen segment] 80 → 3073 [ACK] Seq=2453408091 Ack=2731219739 Win=63925 Len=0

요점:

2개의 캡처는 거의 동일합니다(ISN 임의 지정 고려).

1. TCP 3-way 핸드셰이크가 있습니다.
2. TCP 재전송과 패킷 손실의 징후가 있습니다.
3. Wireshark에 의해 형식이 잘못된 것으로 식별되는 패킷(TCP ACK)이 있습니다.

잘못된 형식의 패킷을 확인합니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909193	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 15:33:31.909849	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=867575959 Ack=4231766829 Win=64240 Len=0 MSS=1380 SACK_PERM=1
3	2013-08-08 15:33:31.913267	192.168.21.193	192.168.14.250	TCP	60	3072 → 80 [ACK] Seq=4231766829 Ack=867575960 Win=65535 Len=2[Malformed Packet]

```

> Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: BelkinIn_63:90:f3 (ec:1a:59:63:90:f3), Dst: Cisco_61:cc:9b (58:8d:09:61:cc:9b)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
> Internet Protocol Version 4, Src: 192.168.21.193, Dst: 192.168.14.250
v Transmission Control Protocol, Src Port: 3072, Dst Port: 80, Seq: 4231766829, Ack: 867575960, Len: 2
  Source Port: 3072
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 2]
  Sequence number: 4231766829
  [Next sequence number: 4231766831]
  Acknowledgment number: 867575960
  0101 ... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window size value: 65535
  [Calculated window size: 65535]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0x01bf [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
  TCP payload (2 bytes)
v [Malformed Packet: Tunnel Socket]
  [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
  [Malformed Packet (Exception occurred)]
  [Severity level: Error]
  [Group: Malformed]
0000 58 8d 09 61 cc 9b ec 1a 59 63 90 f3 81 00 00 14  X..a....Yc.....
0010 08 00 45 00 00 2a 7f 1d 40 00 80 06 d5 a4 c0 a8  ..E...@.....
0020 15 c1 c0 a8 0e fa 0c 00 00 50 fc 3b a7 7d 33 b6  .....P...-3.
0030 28 98 50 10 ff ff 01 bf 00 00 00 00          (-P.....)
  
```

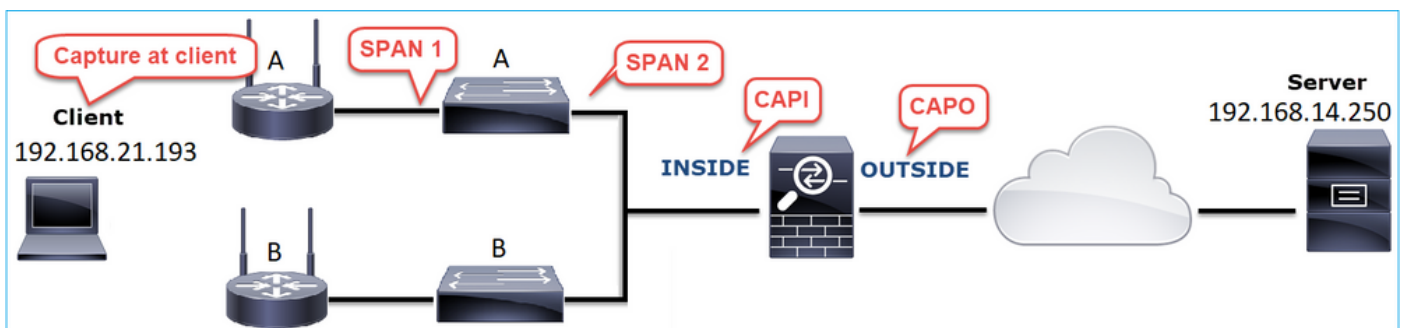
**요점:**

1. 패킷은 Wireshark에 의해 형식이 잘못된 것으로 식별됩니다.
2. 길이는 2바이트입니다.
3. 2바이트의 TCP 페이로드가 있습니다.
4. 페이로드는 4개의 추가 0(00 00)입니다.

**권장 작업**

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. 추가 캡처 수행 엔드포인트에 캡처를 포함하고 가능하면 분할 정복(divide and conquer) 방법을 적용하여 패킷 손상의 소스를 격리합니다. 예를 들면 다음과 같습니다.

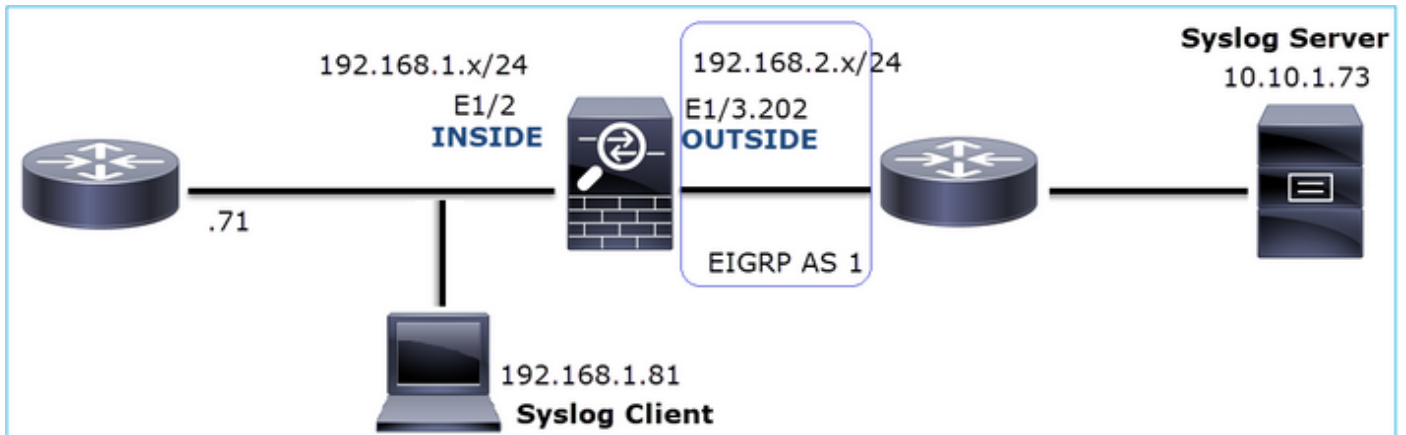


이 경우 스위치 'A' 인터페이스 드라이버에 2바이트가 추가되었으며, 손상이 발생하는 스위치를 교체하는 것이 해결책이었습니다.

## 사례 8. UDP 연결 문제(누락된 패킷)

문제 설명: 대상 Syslog 서버에 Syslog(UDP 514) 메시지가 표시되지 않습니다.

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: 192.168.1.81

Dst IP: 10.10.1.73

프로토콜: UDP 514

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

```
<#root>
```

```
firepower#
```

```
capture CAPI int INSIDE trace match udp host 192.168.1.81 host 10.10.1.73 eq 514
```

```
firepower#
```

```
capture CAPO int OUTSIDE match udp host 192.168.1.81 host 10.10.1.73 eq 514
```

FTD 캡처 시 패킷 표시 안 함:

```
<#root>
```

```
firepower#
```

```
show capture
```

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

```
  match udp host 192.168.1.81 host 10.10.1.73 eq syslog
```

```
capture CAPO type raw-data interface OUTSIDE [Capturing - 0 bytes]
```

```
match udp host 192.168.1.81 host 10.10.1.73 eq syslog
```

## 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. FTD 연결 테이블을 확인합니다.

특정 연결을 확인하려면 다음 구문을 사용할 수 있습니다.

```
<#root>
```

```
firepower#
```

```
show conn address 192.168.1.81 port 514
```

```
10 in use, 3627189 most used
```

```
Inspect Snort:
```

```
    preserve-connection: 6 enabled, 0 in effect, 74 most enabled, 0 most in effect
```

```
UDP
```

```
INSIDE
```

```
10.10.1.73:514
```

```
INSIDE
```

```
192.168.1.81:514, idle 0:00:00, bytes
```

```
480379697
```

```
, flags -
```

```
o
```

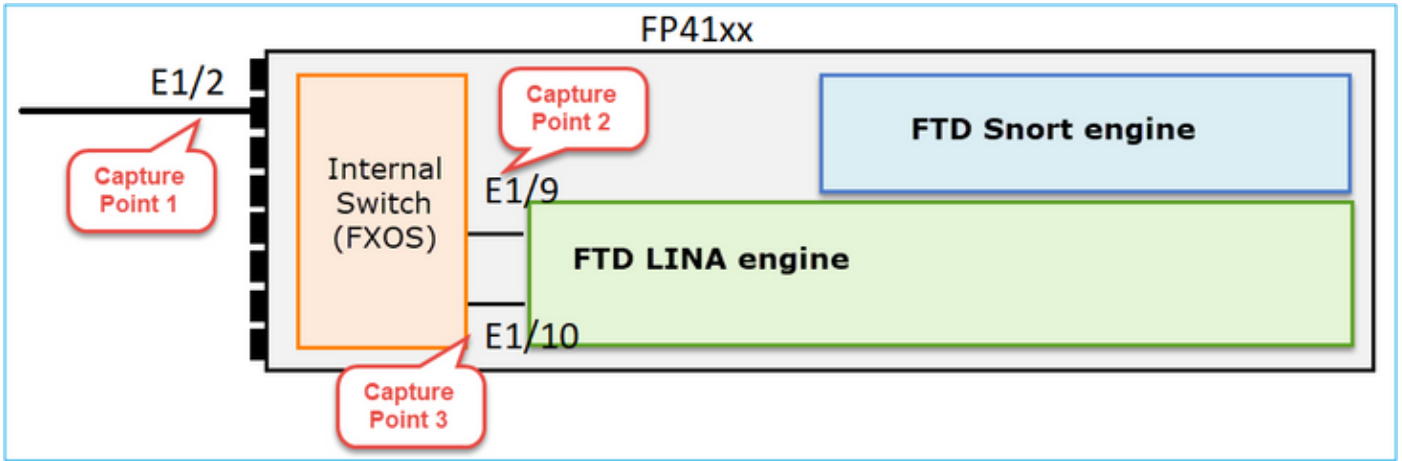
```
N1
```

## 요점:

1. 인그레스(ingress) 및 이그레스(egress) 인터페이스가 동일합니다(U-turn).
2. 바이트 수가 상당히 큰 값(~5GBytes)입니다.
3. 플래그 'o'는 플로우 오프로드(HW accelerated flow)를 의미한다. 따라서 FTD 캡처에서 패킷을 표시하지 않습니다. 플로우 오프로드는 41xx 및 93xx 플랫폼에서만 지원됩니다. 이 경우 디바이스는 41xx입니다.

작업 2. 새시 레벨 캡처를 수행합니다.

이미지에 표시된 대로 Firepower 새시 관리자에 연결하고 인그레스 인터페이스(이 경우 E1/2) 및 백 플레인 인터페이스(E1/9 및 E1/10)에서 캡처를 활성화합니다.



Overview Interfaces Logical Devices Security Engine Platform Settings System Tools Help admin

Select an instance: mzafeiro\_FTD

mzafeiro\_FTD

Ethernet1/2  Ethernet1/3  Ethernet1/1

FTD  
Ethernet1/9, Ethernet1/10

Session Name\* CAPI

Selected Interfaces Ethernet1/2

Buffer Size 256 MB

Snap length: 1518 Bytes

Store Packets **Overwrite** Append

Capture On All Backplane Ports

Capture Filter **Apply Filter** Capture All  
Apply Another Filter Create Filter

몇 초 후:

Capture Session Filter List

CAPI Drop Count: 40103750 Operational State: DOWN - Memory\_Overshoot

Interface Name	Filter	File Size (in bytes)	File Name	Device Name
Ethernet1/10	None	276	CAPI-ethernet-1-10-0.pcap	mzafeiro_FTD
Ethernet1/9	None	132276060	CAPI-ethernet-1-9-0.pcap	mzafeiro_FTD
Ethernet1/2	None	136234072	CAPI-ethernet-1-2-0.pcap	mzafeiro_FTD

🔍 **팁:** Wireshark에서는 VN 태그가 지정된 패킷을 제외하여 물리적 인터페이스 레벨에서 패킷 중복을 제거합니다.

공격 전:



CAPI-ethernet-1-2-0.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
2	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
3	0.0532	Vmware_85:4f:ca	Broadcast	ARP	70	Who has 192.168.103.111? Tell 192.168.103.112
4	0.0000	Vmware_85:4f:ca	Broadcast	ARP	64	Who has 192.168.103.111? Tell 192.168.103.112
5	0.5216	Vmware_85:2f:00	Broadcast	ARP	70	Who has 10.10.10.1? Tell 10.10.10.10
6	0.0000	Vmware_85:2f:00	Broadcast	ARP	64	Who has 10.10.10.1? Tell 10.10.10.10
7	0.5770	Vmware_85:2f:00	Broadcast	ARP	70	Who has 10.10.10.1? Tell 10.10.10.10
8	0.0000	Vmware_85:2f:00	Broadcast	ARP	64	Who has 10.10.10.1? Tell 10.10.10.10
9	0.8479	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
10	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
11	0.1520	Vmware_85:2f:00	Broadcast	ARP	70	Who has 10.10.10.1? Tell 10.10.10.10
12	0.0000	Vmware_85:2f:00	Broadcast	ARP	64	Who has 10.10.10.1? Tell 10.10.10.10
13	0.8606	Vmware_85:4f:ca	Broadcast	ARP	70	Who has 192.168.103.111? Tell 192.168.103.112
14	0.0000	Vmware_85:4f:ca	Broadcast	ARP	64	Who has 192.168.103.111? Tell 192.168.103.112
15	0.1655	192.168.0.101	173.38.200.100	DNS	91	Standard query 0x4a9f A 2.debian.pool.ntp.org
16	0.0000	192.168.0.101	173.38.200.100	DNS	85	Standard query 0x4a9f A 2.debian.pool.ntp.org
17	0.0000	192.168.0.101	173.38.200.100	DNS	91	Standard query 0x4afd AAAA 2.debian.pool.ntp.org
18	0.0000	192.168.0.101	173.38.200.100	DNS	85	Standard query 0x4afd AAAA 2.debian.pool.ntp.org
19	0.0003	192.168.0.101	173.38.200.100	DNS	91	Standard query 0x4a9f A 2.debian.pool.ntp.org
20	0.0000	192.168.0.101	173.38.200.100	DNS	85	Standard query 0x4a9f A 2.debian.pool.ntp.org

이후:

CAPI-ethernet-1-2-0.pcap

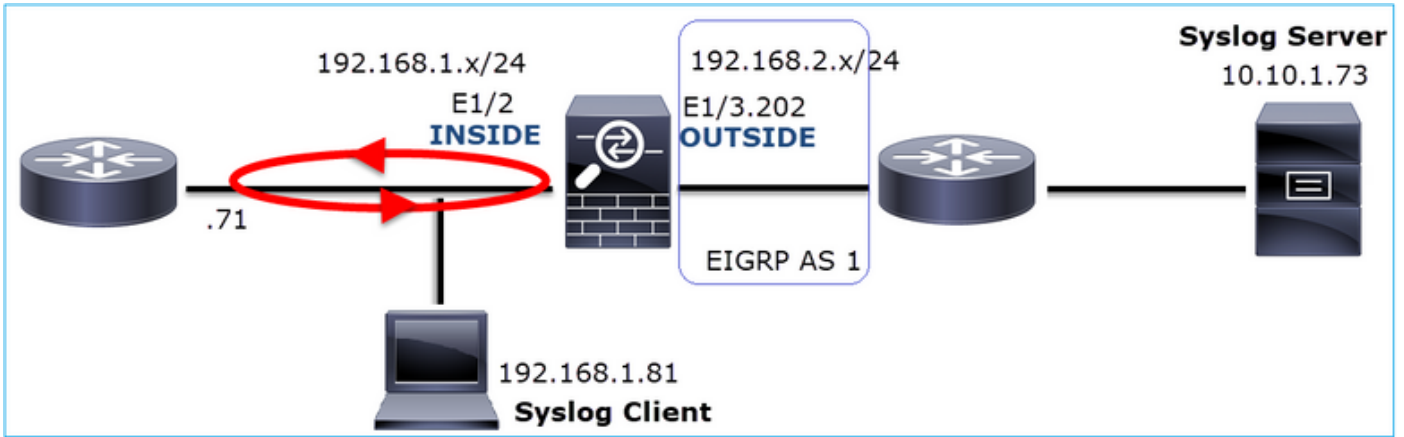
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

syslog && !mtag

No.	Time	Source	Destination	Protocol	Length	Time to live	Info
1334	0.000000000	192.168.1.81	10.10.1.73	Syslog	147		255 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1336	0.00078873	192.168.1.81	10.10.1.73	Syslog	147		254 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1338	0.00015099	192.168.1.81	10.10.1.73	Syslog	147		253 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1340	0.000128919	192.168.1.81	10.10.1.73	Syslog	131		255 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1342	0.000002839	192.168.1.81	10.10.1.73	Syslog	147		252 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1344	0.000137974	192.168.1.81	10.10.1.73	Syslog	131		254 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1346	0.000002758	192.168.1.81	10.10.1.73	Syslog	147		251 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1348	0.000261845	192.168.1.81	10.10.1.73	Syslog	131		253 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1350	0.000002736	192.168.1.81	10.10.1.73	Syslog	147		250 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1352	0.000798149	192.168.1.81	10.10.1.73	Syslog	200		255 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1354	0.000498621	192.168.1.81	10.10.1.73	Syslog	131		252 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1356	0.000002689	192.168.1.81	10.10.1.73	Syslog	147		249 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1358	0.000697783	192.168.1.81	10.10.1.73	Syslog	195		255 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
1360	0.000599702	192.168.1.81	10.10.1.73	Syslog	151		255 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
1362	0.000002728	192.168.1.81	10.10.1.73	Syslog	200		254 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1364	0.000499914	192.168.1.81	10.10.1.73	Syslog	131		251 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1366	0.000697761	192.168.1.81	10.10.1.73	Syslog	147		248 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1368	0.000169137	192.168.1.81	10.10.1.73	Syslog	195		254 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
1370	0.000433196	192.168.1.81	10.10.1.73	Syslog	151		254 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
1372	0.000498718	192.168.1.81	10.10.1.73	Syslog	200		253 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1374	0.000002849	192.168.1.81	10.10.1.73	Syslog	131		250 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1376	0.000596345	192.168.1.81	10.10.1.73	Syslog	147		247 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1378	0.000600157	192.168.1.81	10.10.1.73	Syslog	195		253 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
1380	0.000002772	192.168.1.81	10.10.1.73	Syslog	151		253 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
1382	0.000600947	192.168.1.81	10.10.1.73	Syslog	200		252 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1384	0.000498808	192.168.1.81	10.10.1.73	Syslog	131		249 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n

요점:

1. 표시 필터는 패킷 중복을 제거하고 syslog만 표시하는 데 적용됩니다.
2. 패킷 간의 차이는 마이크로초 수준입니다. 이는 매우 높은 패킷 속도를 나타냅니다.
3. TTL(Time to Live) 값이 지속적으로 감소합니다. 패킷 루프를 나타냅니다.



작업 3. 패킷 추적기를 사용합니다.

패킷이 방화벽 LINA 엔진을 통과하지 않으므로 라이브 추적(캡처 w/trace)은 수행할 수 없지만 패킷 추적기를 사용하여 에뮬레이트된 패킷을 추적할 수 있습니다.

```
<#root>
```

```
firepower#
```

```
packet-tracer input INSIDE udp 10.10.1.73 514 192.168.1.81 514
```

```
Phase: 1
```

```
Type: CAPTURE
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
MAC Access list
```

```
Phase: 2
```

```
Type: ACCESS-LIST
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Implicit Rule
```

```
Additional Information:
```

```
MAC Access list
```

```
Phase: 3
```

```
Type: FLOW-LOOKUP
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Found flow with id 25350892, using existing flow
```

```
Phase: 4
```

```
Type: SNORT
```

```
Subtype:
```

```
Result: ALLOW
```

```
Config:
```

```
Additional Information:
```

```
Snort Verdict: (fast-forward) fast forward this flow
```

```
Phase: 5
```

Type: ROUTE-LOOKUP  
Subtype: Resolve Egress Interface  
Result: ALLOW  
Config:  
Additional Information:  
found next-hop 192.168.1.81 using egress ifc INSIDE

Phase: 6  
Type: ADJACENCY-LOOKUP  
Subtype: next-hop and adjacency  
Result: ALLOW  
Config:  
Additional Information:  
adjacency Active  
next-hop mac address a023.9f92.2a4d hits 1 reference 1

Phase: 7  
Type: CAPTURE  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
MAC Access list

Result:

**input-interface: INSIDE**

input-status: up

input-line-status: up

**output-interface: INSIDE**

output-status: up

output-line-status: up

Action: allow

작업 4. FTD 라우팅을 확인합니다.

방화벽 라우팅 테이블에서 라우팅 문제가 있는지 확인합니다.

<#root>

firepower#

show route 10.10.1.73

Routing entry for 10.10.1.0 255.255.255.0

Known via "eigrp 1", distance 90, metric 3072, type internal

Redistributing via eigrp 1

Last update from 192.168.2.72 on

OUTSIDE, 0:03:37 ago

Routing Descriptor Blocks:

\* 192.168.2.72, from 192.168.2.72,

0:02:37 ago, via OUTSIDE

Route metric is 3072, traffic share count is 1

Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit  
Reliability 255/255, minimum MTU 1500 bytes  
Loading 29/255, Hops 1

요점:

1. 경로가 올바른 이그레스 인터페이스를 가리킵니다.
2. 경로는 몇 분 전에 학습되었습니다(0:02:37).

작업 5. 연결 가동 시간을 확인합니다.

연결 업타임을 확인하여 이 연결이 설정된 시간을 확인합니다.

<#root>

firepower#

```
show conn address 192.168.1.81 port 514 detail
```

21 in use, 3627189 most used

Inspect Snort:

preserve-connection: 19 enabled, 0 in effect, 74 most enabled, 0 most in effect

Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,

b - TCP state-bypass or nailed,

C - CTIQBE media, c - cluster centralized,

D - DNS, d - dump, E - outside back connection, e - semi-distributed,

F - initiator FIN, f - responder FIN,

G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,

i - incomplete, J - GTP, j - GTP data, K - GTP t3-response

k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media

N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in effect)

n - GUP, O - responder data, o - offloaded,

P - inside back connection, p - passenger flow

q - SQL\*Net data, R - initiator acknowledged FIN,

R - UDP SUNRPC, r - responder acknowledged FIN,

T - SIP, t - SIP transient, U - up,

V - VPN orphan, v - M3UA W - WAAS,

w - secondary domain backup,

X - inspected by service module,

x - per session, Y - director stub flow, y - backup stub flow,

Z - Scansafe redirection, z - forwarding stub flow

UDP INSIDE: 10.10.1.73/514 INSIDE: 192.168.1.81/514,  
flags -oN1, idle 0s,

uptime 3m49s

, timeout 2m0s, bytes 4801148711

요점:

1. 4분 전에 연결이 설정되었습니다(라우팅 테이블에서 EIGRP 경로 설치 전).

작업 6. 설정된 연결을 지웁니다.

이 경우 패킷은 설정된 연결과 일치하며 잘못된 이그레스 인터페이스로 라우팅됩니다. 이로 인해 루프가 발생합니다. 이는 방화벽 운영 순서 때문입니다.

1. 설정된 연결 조회(전역 라우팅 테이블 조회에 우선함)
2. NAT(Network Address Translation) 조회 - UN-NAT(destination NAT) 단계가 PBR 및 경로 조회보다 우선합니다.
3. PBR(Policy-Based Routing)
4. 전역 라우팅 테이블 조회

연결이 시간 초과되지 않으므로(UDP conn 유희 시간 제한이 2분인 동안 Syslog 클라이언트가 지속적으로 패킷을 전송함) 연결을 수동으로 지워야 합니다.

<#root>

firepower#

```
clear conn address 10.10.1.73 address 192.168.1.81 protocol udp port 514
```

1 connection(s) deleted.

새 연결이 설정되었는지 확인합니다.

<#root>

firepower#

```
show conn address 192.168.1.81 port 514 detail | b 10.10.1.73.*192.168.1.81
```

UDP

OUTSIDE

: 10.10.1.73/514

INSIDE

: 192.168.1.81/514,  
flags -oN1, idle 1m15s, uptime 1m15s, timeout 2m0s, bytes 408

작업 7. 부동 연결 시간 제한을 구성합니다.

이는 문제를 해결하고 특히 UDP 플로우의 경우 최적화되지 않은 라우팅을 방지하기 위한 적절한 솔루션입니다. Devices(디바이스) > Platform Settings(플랫폼 설정) > Timeouts(시간 제한)로 이동하여 값을 설정합니다.

SMTP Server	H.323	Default	0:05:00	(0:0:0 or 0:0:0 - 1193:0:0)
SNMP	SIP	Default	0:30:00	(0:0:0 or 0:5:0 - 1193:0:0)
SSL	SIP Media	Default	0:02:00	(0:0:0 or 0:1:0 - 1193:0:0)
Syslog	SIP Disconnect:	Default	0:02:00	(0:02:0 or 0:0:1 - 0:10:0)
Timeouts	SIP Invite	Default	0:03:00	(0:1:0 or 0:1:0 - 0:30:0)
Time Synchronization	SIP Provisional Media	Default	0:02:00	(0:2:0 or 0:1:0 - 0:30:0)
UCAPL/CC Compliance	Floating Connection	Custom	0:00:30	(0:0:0 or 0:0:30 - 1193:0:0)
	Xlate-PAT	Default	0:00:30	(0:0:30 or 0:0:30 - 0:5:0)

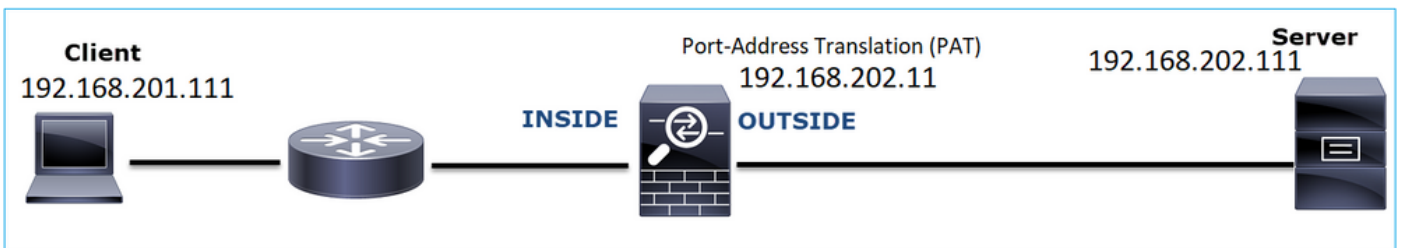
부동 연결 시간 제한에 대한 자세한 내용은 명령 참조에서 확인할 수 있습니다.

<https://www.cisco.com/c/en/us/td/docs/security/asa/asa-command-reference/T-Z/cmdref4/t1.html#pgfid-1649892>

## 사례 9. HTTPS 연결 문제(시나리오 1)

문제 설명: 클라이언트 192.168.201.105와 서버 192.168.202.101 간의 HTTPS 통신을 설정할 수 없습니다.

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: 192.168.201.111

Dst IP: 192.168.202.111

프로토콜: TCP 443(HTTPS)

캡처 분석

FTD LINA 엔진에서 캡처를 활성화합니다.

OUTSIDE 캡처에 사용되는 IP는 Port-Address Translation 컨피그레이션으로 인해 달라집니다.

```
<#root>
firepower#
capture CAPI int INSIDE match ip host 192.168.201.111 host 192.168.202.111
firepower#
capture CAPO int OUTSIDE match ip host 192.168.202.11 host 192.168.202.111
```

이 이미지는 NGFW INSIDE 인터페이스에서 캡처한 내용을 보여줍니다.

No.	Time	Source	Destination	Protocol	Length	Identification	Info
38	2018-02-01 10:39:35.187887	192.168.201.111	192.168.202.111	TCP	78	0x2f31 (12081)	6666 → 443 [SYN] Seq=2034865631 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=192658158 TSecr=0 WS=128
39	2018-02-01 10:39:35.188999	192.168.202.111	192.168.201.111	TCP	78	0x0900 (0)	443 → 6666 [SYN, ACK] Seq=4086514531 Ack=2034865632 Win=28960 Len=0 MSS=1380 SACK_PERM=1 TSval=3119
40	2018-02-01 10:39:35.189046	192.168.201.111	192.168.202.111	TCP	70	0x2f32 (12082)	6666 → 443 [ACK] Seq=2034865632 Ack=4086514532 Win=29312 Len=0 TSval=192658158 TSecr=3119615816
41	2018-02-01 10:39:35.251695	192.168.201.111	192.168.202.111	TLSv1	326	0x2f33 (12083)	Client Hello
42	2018-02-01 10:39:35.252352	192.168.202.111	192.168.201.111	TCP	70	0xefb4 (61364)	443 → 6666 [ACK] Seq=4086514532 Ack=2034865888 Win=8192 Len=0 TSval=3119615816 TSecr=192658174
43	2018-02-01 10:40:05.317320	192.168.202.111	192.168.201.111	TCP	70	0xd8c3 (55491)	443 → 6666 [RST] Seq=4086514532 Win=8192 Len=0 TSval=3119645908 TSecr=0

요점:

1. TCP 3-way 핸드셰이크가 있습니다.
2. SSL 협상이 시작됩니다. 클라이언트가 Client Hello 메시지를 보냅니다.
3. 클라이언트로 전송된 TCP ACK가 있습니다.
4. 클라이언트로 전송된 TCP RST가 있습니다.

이 그림에서는 NGFW OUTSIDE 인터페이스에서 캡처한 내용을 보여줍니다.

No.	Time	Source	Destination	Protocol	Length	Identification	Info
33	2018-02-01 10:39:35.188192	192.168.202.11	192.168.202.111	TCP	78	0x2f31 (12081)	15880 → 443 [SYN] Seq=2486930707 Win=29200 Len=0 MSS=1380 SACK_PERM=1 TSval=192658158 TSecr=0 WS=128
34	2018-02-01 10:39:35.188527	192.168.202.111	192.168.202.11	TCP	78	0x0900 (0)	443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=19
35	2018-02-01 10:39:35.189214	192.168.202.11	192.168.202.111	TCP	70	0x2f32 (12082)	15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Win=29312 Len=0 TSval=192658158 TSecr=3119615816
36	2018-02-01 10:39:35.252397	192.168.202.11	192.168.202.111	TLSv1	257	0xcd36 (52534)	Client Hello
37	2018-02-01 10:39:37.274430	192.168.202.11	192.168.202.111	TCP	257	0xb905 (47365)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192660198 TSecr=0
38	2018-02-01 10:39:41.297332	192.168.202.11	192.168.202.111	TCP	257	0x88af (34991)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192664224 TSecr=0
39	2018-02-01 10:39:49.309569	192.168.202.11	192.168.202.111	TCP	257	0xf68a (63114)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192672244 TSecr=0
40	2018-02-01 10:40:05.317305	192.168.202.11	192.168.202.111	TCP	70	0xd621 (54817)	15880 → 443 [RST] Seq=2486930895 Win=8192 Len=0 TSval=192688266 TSecr=0
41	2018-02-01 10:40:06.798700	192.168.202.111	192.168.202.11	TCP	78	0x0900 (0)	[TCP Retransmission] 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSv

요점:

1. TCP 3-way 핸드셰이크가 있습니다.
2. SSL 협상이 시작됩니다. 클라이언트가 Client Hello 메시지를 보냅니다.
3. 방화벽에서 서버로 전송되는 TCP 재전송이 있습니다.
4. 서버로 전송된 TCP RST가 있습니다.

### 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 줄이기 위한 목적입니다.

#### 작업 1. 추가 캡처 수행

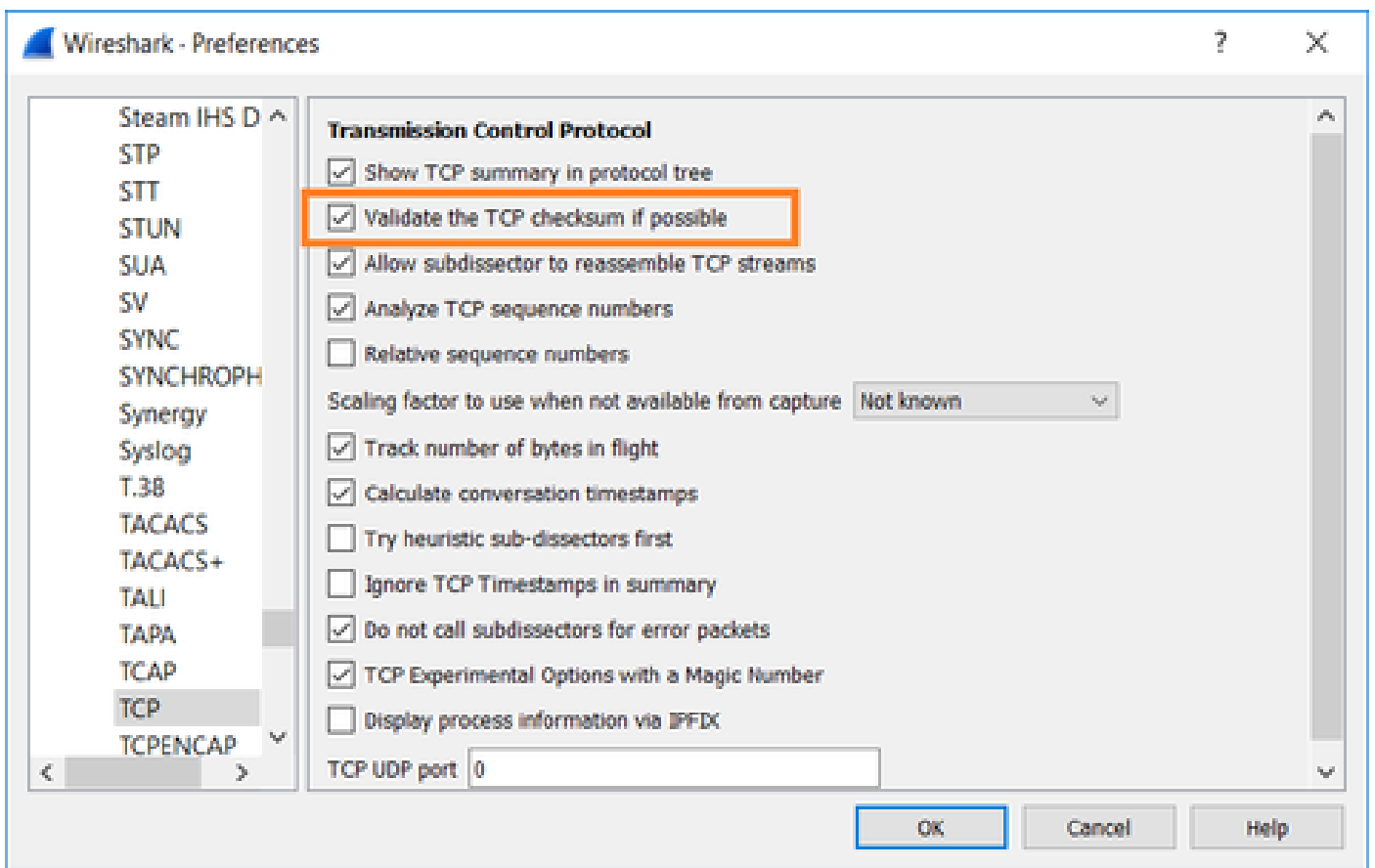
서버에서 캡처한 결과, 서버가 손상된 TCP 체크섬과 함께 TLS 클라이언트 Hello를 수신한 다음 이

를 자동으로 삭제합니다(클라이언트에 대한 TCP RST 또는 기타 응답 패킷이 없음).

```
21:26:27.133677 IP (tos 0x0, ttl 64, id 52534, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x0c65 (incorrect -> 0x3063), seq 1:188, ack 1, win 64, options [nop,nop,T
  S val 192658174 ecr 3119615816], length 187
21:26:29.155652 IP (tos 0x0, ttl 64, id 47365, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x4db7 (incorrect -> 0x71b5), seq 1:188, ack 1, win 64, options [nop,nop,T
  S val 192660198 ecr 0], length 187
21:26:33.178142 IP (tos 0x0, ttl 64, id 34991, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x3d d (incorrect -> 0x61fb), seq 1:188, ack 1, win 64, options [nop,nop,T
  S val 192664224 ecr 0], length 187
21:26:41.189640 IP (tos 0x0, ttl 64, id 63114, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x1e9 (incorrect -> 0x42a7), seq 1:188, ack 1, win 64, options [nop,nop,T
  S val 192672244 ecr 0], length 187
21:26:57.195947 IP (tos 0x0, ttl 64, id 54817, offset 0, flags [DF], proto TCP (6), length 52)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [R], cksum 0x9ee (incorrect -> 0xc2e8), seq 2486930895, win 64, options [nop,nop,TS v
  al 192688266 ecr 0], length 0
21:26:58.668973 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
  192.168.202.111.443 > 192.168.202.11.15880: Flags [S.], cksum 0x15fb (incorrect -> 0xffd2), seq 3674405382, ack 2486930708, win 28960, o
  ptions [mss 1460,sackOK,TS val 3119647415 ecr 192658158,nop,wscale 7], length 0
^C
154 packets captured
154 packets received by filter
```

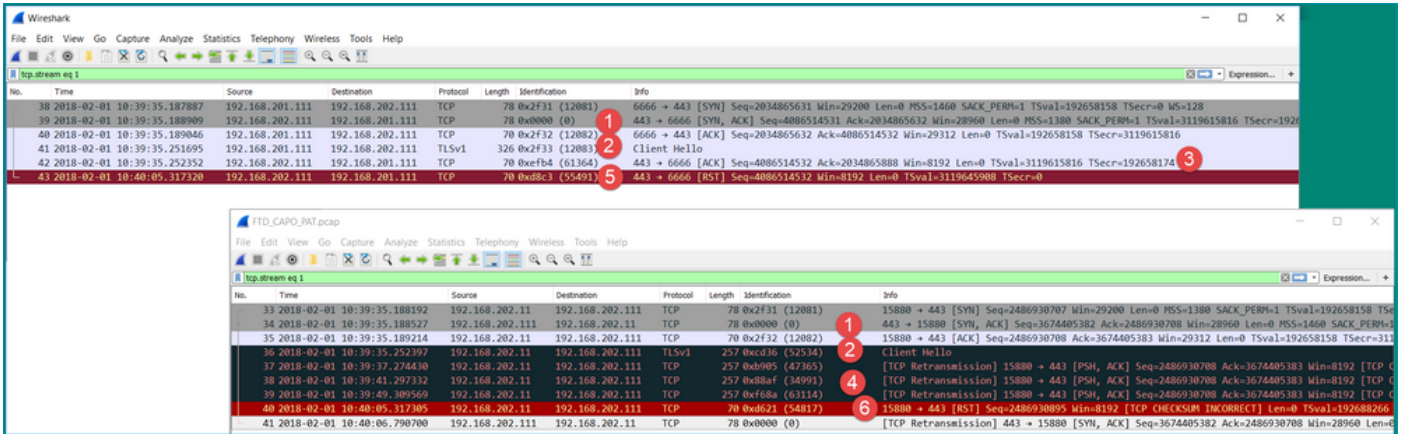
모든 것을 종합할 때

이 경우 Wireshark에서 Validate the TCP checksum if possible(TCP 체크섬 검증) 옵션을 활성화해  
야 합니다. 이미지에 표시된 대로 Edit(편집) > Preferences(환경 설정) > Protocols(프로토콜) >  
TCP로 이동합니다.



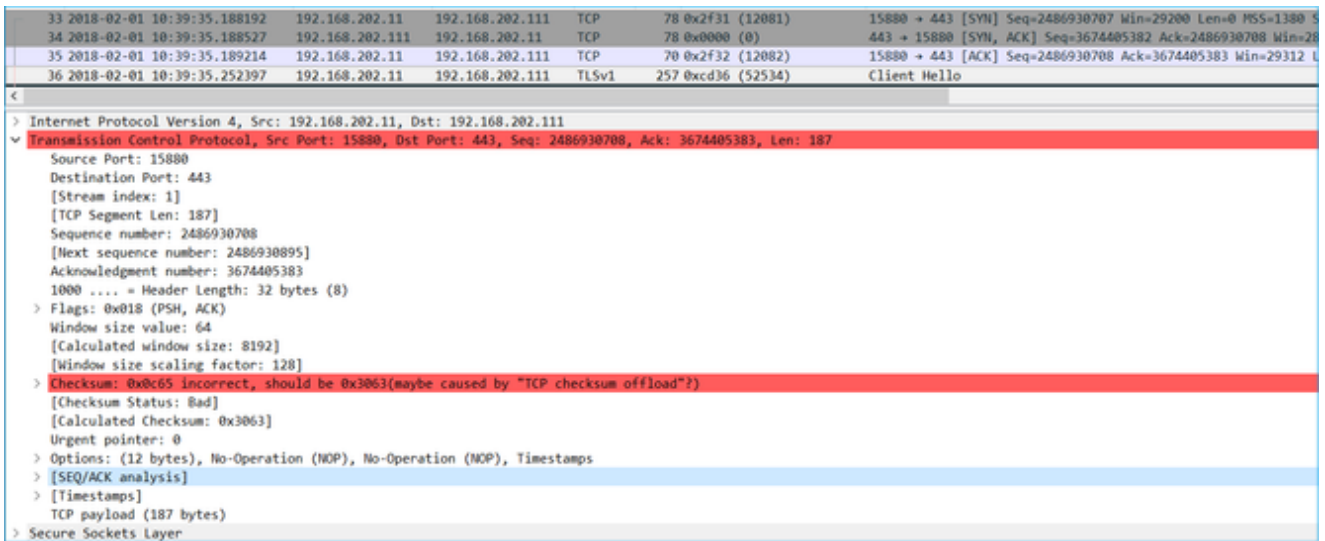
이 경우 전체 그림을 보려면 캡처를 나란히 배치하면 유용합니다.





**요점:**

1. TCP 3-way 핸드셰이크가 있습니다. IP ID가 동일합니다. 이는 플로우가 방화벽에 의해 프록시되지 않았음을 의미합니다.
2. TLS Client Hello는 IP ID가 12083인 클라이언트에서 제공됩니다. 패킷은 방화벽에 의해 프록시되고(이 경우 방화벽은 TLS 암호 해독 정책으로 구성됨) IP ID가 52534으로 변경됩니다. 또한 나중에 수정되는 소프트웨어 결함으로 인해 패킷 TCP 체크섬이 손상됩니다.
3. 방화벽은 TCP 프록시 모드에 있으며, (서버를 스푸핑하는) 클라이언트에 ACK를 전송합니다.



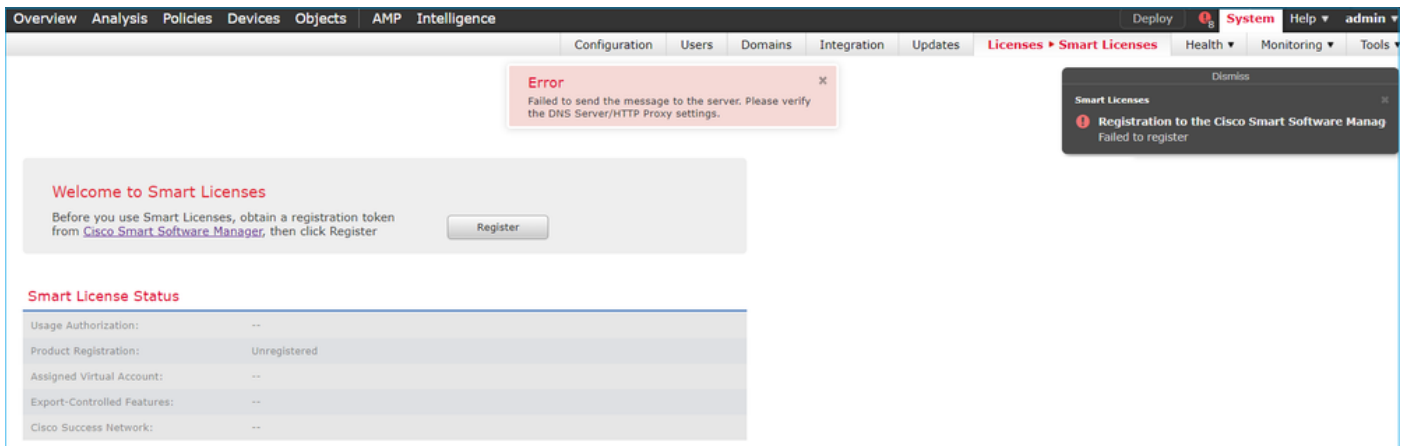
4. 방화벽은 서버에서 TCP ACK 패킷을 수신하지 않고 TLS Client Hello 메시지를 다시 전송합니다. 이는 방화벽이 활성화한 TCP 프록시 모드 때문입니다.
5. ~30초 후 방화벽이 중단되고 클라이언트로 TCP RST를 전송합니다.
6. 방화벽은 서버를 향해 TCP RST를 전송합니다.

**참조:**

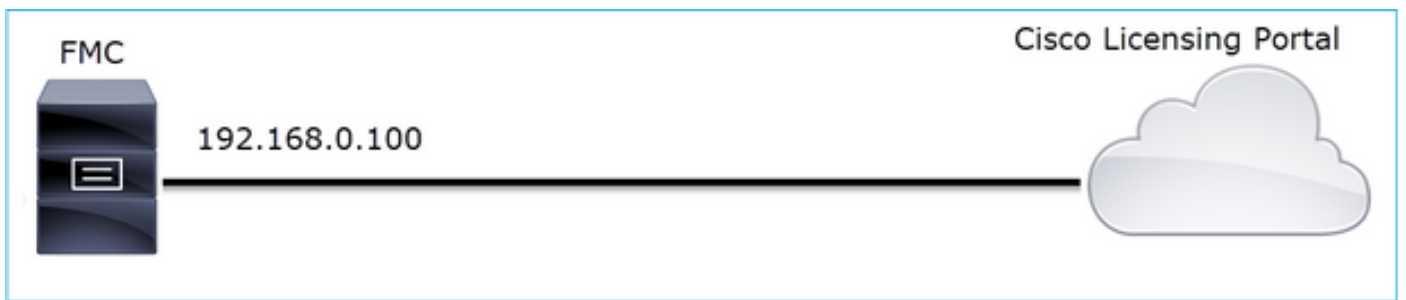
[Firepower TLS/SSL 핸드셰이크 처리](#)

## 사례 10. HTTPS 연결 문제(시나리오 2)

문제 설명: FMC Smart License 등록이 실패합니다.



이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

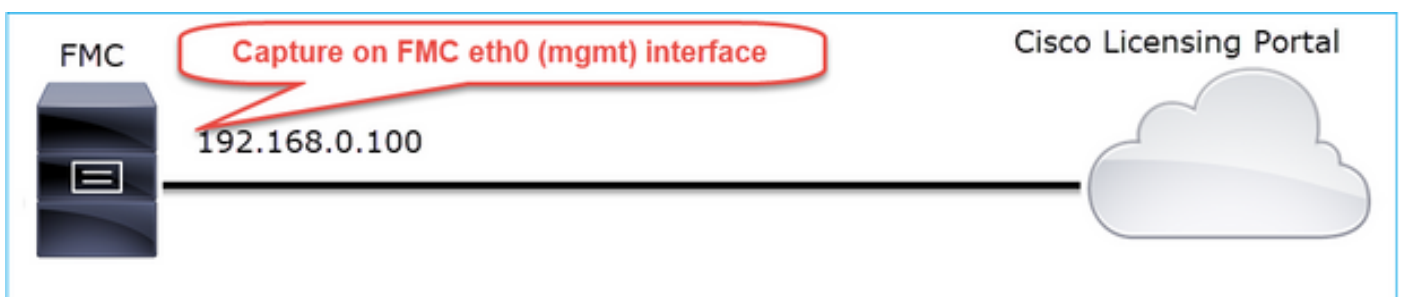
소스 IP: 192.168.0.100

Dst: tools.cisco.com

프로토콜: TCP 443(HTTPS)

캡처 분석

FMC 관리 인터페이스에서 캡처를 활성화합니다.



다시 등록해 보십시오. Error(오류) 메시지가 나타나면 Ctrl-C를 눌러 캡처를 중지합니다.

<#root>

root@firepower:/Volume/home/admin#

tcpdump -i eth0 port 443 -s 0 -w CAP.pcap

HS\_PACKET\_BUFFER\_SIZE is set to 4.

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

^C

264 packets captured

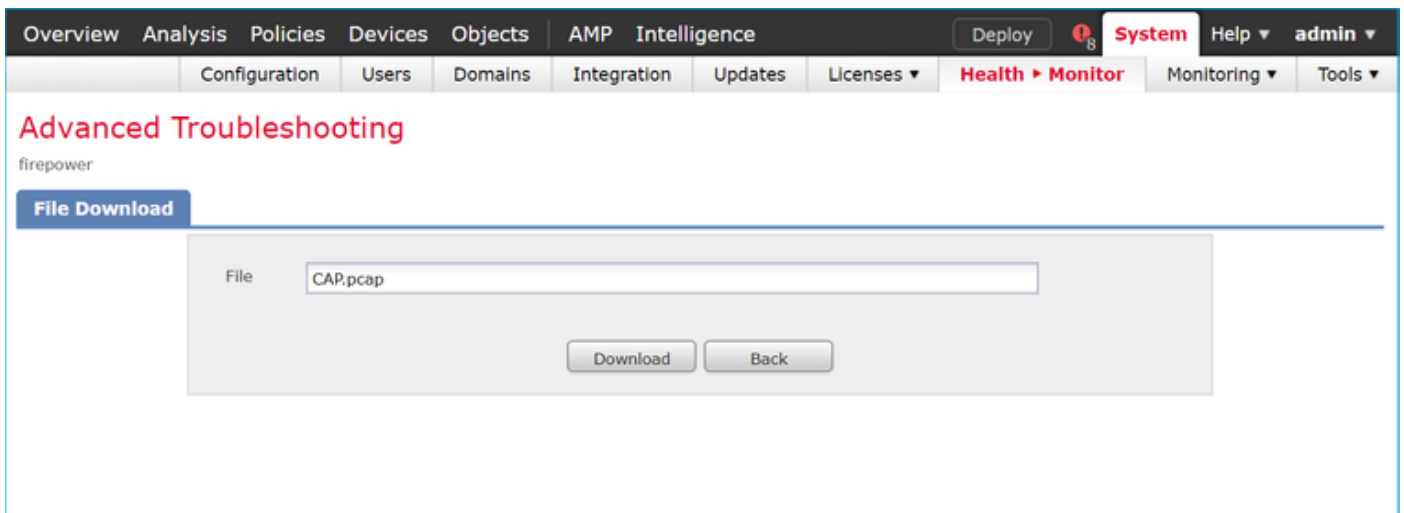
<- CTRL-C

264 packets received by filter

0 packets dropped by kernel

root@firepower:/Volume/home/admin#

이미지에 표시된 대로 FMC에서 캡처를 수집합니다(System > Health > Monitor, 디바이스 선택 및 Advanced Troubleshooting 선택).



이 그림에서는 Wireshark의 FMC 캡처를 보여 줍니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-23 07:44:59.218797	192.168.0.100	10.229.20.96	TLSv1.2	107	Application Data
2	2019-10-23 07:44:59.220929	10.229.20.96	192.168.0.100	TLSv1.2	123	Application Data
3	2019-10-23 07:44:59.220960	192.168.0.100	10.229.20.96	TCP	54	443 → 64722 [ACK] Seq=1380971613 Ack=2615750168 Win=249 Len=0
4	2019-10-23 07:45:02.215376	192.168.0.100	10.229.20.96	TLSv1.2	107	Application Data
5	2019-10-23 07:45:02.217321	10.229.20.96	192.168.0.100	TLSv1.2	123	Application Data
6	2019-10-23 07:45:02.217336	192.168.0.100	10.229.20.96	TCP	54	443 → 64722 [ACK] Seq=1380971666 Ack=2615750237 Win=249 Len=0
7	2019-10-23 07:45:05.215460	192.168.0.100	10.229.20.96	TLSv1.2	107	Application Data
8	2019-10-23 07:45:05.217331	10.229.20.96	192.168.0.100	TLSv1.2	123	Application Data
9	2019-10-23 07:45:05.217345	192.168.0.100	10.229.20.96	TCP	54	443 → 64722 [ACK] Seq=1380971719 Ack=2615750306 Win=249 Len=0
10	2019-10-23 07:45:06.216584	10.229.20.96	192.168.0.100	TCP	66	64784 → 443 [SYN] Seq=4002690284 Win=64240 Len=0 MSS=1380 WS=256 S
11	2019-10-23 07:45:06.216631	192.168.0.100	10.229.20.96	TCP	66	443 → 64784 [SYN, ACK] Seq=3428959426 Ack=4002690285 Win=29200 Len=0
12	2019-10-23 07:45:06.218550	10.229.20.96	192.168.0.100	TCP	60	64784 → 443 [ACK] Seq=4002690285 Ack=3428959427 Win=66048 Len=0
13	2019-10-23 07:45:06.219386	10.229.20.96	192.168.0.100	TLSv1.2	571	Client Hello

🔍 **팁:** 캡처된 모든 새 TCP 세션을 확인하려면 Wireshark에서 tcp.flags==0x2 디스플레이 필터를

🔍 사용합니다. 이렇게 하면 캡처된 모든 TCP SYN 패킷이 필터링됩니다.

CAP.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags==0x2

No.	Time	Source	Destination	Protocol	Length	Info
10	2019-10-23 07:45:06.216584	10.229.20.96	192.168.0.100	TCP	66	64784 → 443 [SYN] Seq=4002690284 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
19	2019-10-23 07:45:06.225743	10.229.20.96	192.168.0.100	TCP	66	64785 → 443 [SYN] Seq=3970528579 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
45	2019-10-23 07:45:12.403280	10.229.20.96	192.168.0.100	TCP	66	64790 → 443 [SYN] Seq=442965162 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
51	2019-10-23 07:45:12.409842	10.229.20.96	192.168.0.100	TCP	66	64791 → 443 [SYN] Seq=77539654 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74	35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16127801 TSecr=0 WS=128
108	2019-10-23 07:45:24.969622	192.168.0.100	72.163.4.38	TCP	74	35756 → 443 [SYN] Seq=1993860949 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16138303 TSecr=0 WS=128
137	2019-10-23 07:45:35.469403	192.168.0.100	173.37.145.8	TCP	74	58326 → 443 [SYN] Seq=723413997 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2040670996 TSecr=0 WS=128
163	2019-10-23 07:45:45.969384	192.168.0.100	173.37.145.8	TCP	74	58330 → 443 [SYN] Seq=2299582550 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2040681496 TSecr=0 WS=128
192	2019-10-23 07:45:56.468604	192.168.0.100	72.163.4.38	TCP	74	35768 → 443 [SYN] Seq=1199682453 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16169802 TSecr=0 WS=128
227	2019-10-23 07:46:07.218984	10.229.20.96	192.168.0.100	TCP	66	64811 → 443 [SYN] Seq=1496581075 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
236	2019-10-23 07:46:07.225881	10.229.20.96	192.168.0.100	TCP	66	64812 → 443 [SYN] Seq=563292608 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1

🔍 팁: SSL Client Hello의 Server Name(서버 이름) 필드를 열로 적용합니다.

75 2019-10-23 07:45:14.634091 192.168.0.100 72.163.4.38 TLSv1.2 571 Client Hello

> Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits)

> Ethernet II, Src: Vmware\_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco\_f6:1d:ae (00:be:75:f6:1d:ae)

> Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38

> Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 517

Secure Sockets Layer


- TLsv1.2 Record Layer: Handshake Protocol
  - Content Type: Handshake (22)
  - Version: TLS 1.0 (0x0301)
  - Length: 512
  - Handshake Protocol: Client Hello
    - Handshake Type: Client Hello (1)
    - Length: 508
    - Version: TLS 1.2 (0x0303)
    - Random: 234490a107438c73b595646532
    - Session ID Length: 0
    - Cipher Suites Length: 100
    - Cipher Suites (50 suites)
    - Compression Methods Length: 1
    - Compression Methods (1 method)
    - Extensions Length: 367
    - Extension: server\_name (len=20)
      - Type: server\_name (0)
      - Length: 20
      - Server Name Indication extension
        - Server Name list length: 18
        - Server Name Type: host\_name (0)
        - Server Name length: 15
        - Server Name: tools.cisco.com

Expand Subtrees  
Collapse Subtrees  
Expand All  
Collapse All  
Apply as Column  
Apply as Filter  
Prepare a Filter  
Conversation Filter  
Colorize with Filter  
Follow  
Copy  
Show Packet Bytes...  
Export Packet Bytes...  
Wiki Protocol Page  
Filter Field Reference  
Protocol Preferences  
Decode As...  
Go to Linked Packet  
Show Linked Packet in New Window

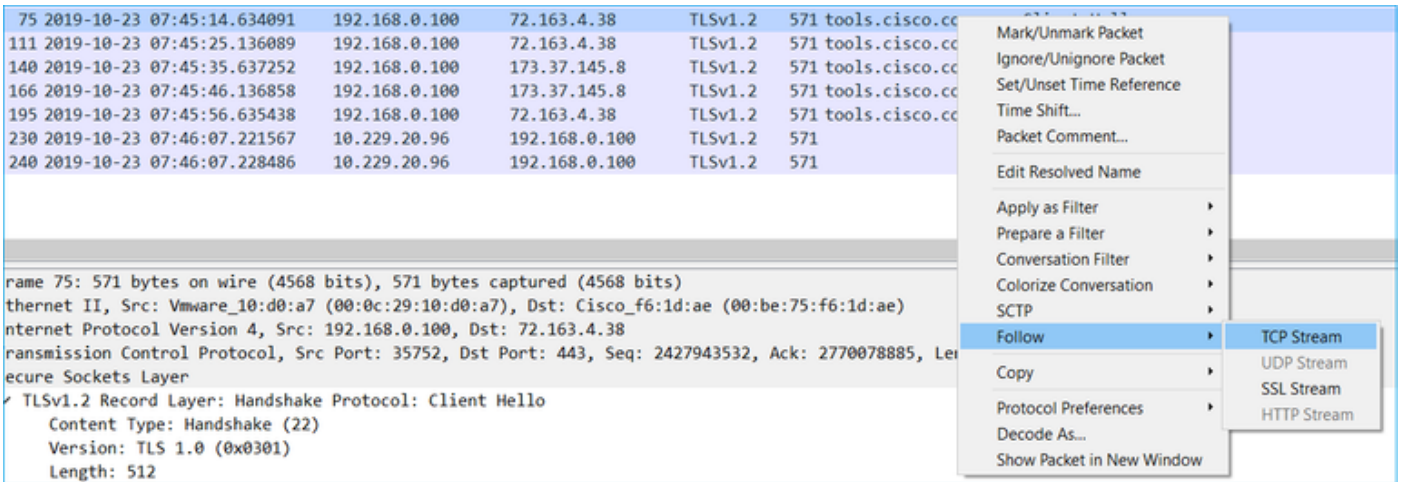
🔍 팁: 이 표시 필터를 적용하여 Client Hello messages ssl.handshake.type == 1만 표시합니다.

ssl.handshake.type == 1

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
13	2019-10-23 07:45:06.219386	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
23	2019-10-23 07:45:06.227250	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
48	2019-10-23 07:45:12.406366	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
54	2019-10-23 07:45:12.412199	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
111	2019-10-23 07:45:25.136089	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
140	2019-10-23 07:45:35.637252	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
166	2019-10-23 07:45:46.136858	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
195	2019-10-23 07:45:56.635438	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
230	2019-10-23 07:46:07.221567	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
240	2019-10-23 07:46:07.228486	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello

 참고: 이 문서를 작성할 때 Smart Licensing 포털(tools.cisco.com)에서는 72.163.4.38, 173.37.145.8의 IP를 사용합니다.

이미지에 표시된 대로 TCP 흐름 중 하나를 따릅니다(Follow > TCP Stream).



75 2019-10-23 07:45:14.634091 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.com

111 2019-10-23 07:45:25.136089 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.com

140 2019-10-23 07:45:35.637252 192.168.0.100 173.37.145.8 TLSv1.2 571 tools.cisco.com

166 2019-10-23 07:45:46.136858 192.168.0.100 173.37.145.8 TLSv1.2 571 tools.cisco.com

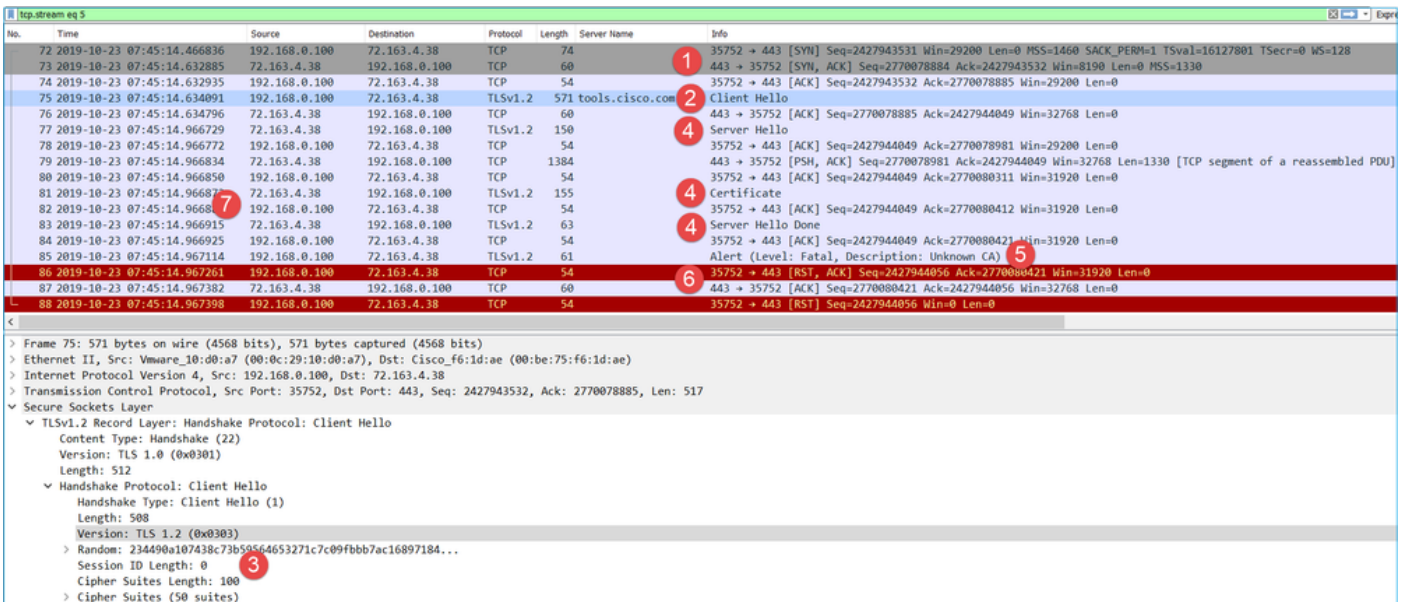
195 2019-10-23 07:45:56.635438 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.com

230 2019-10-23 07:46:07.221567 10.229.20.96 192.168.0.100 TLSv1.2 571

240 2019-10-23 07:46:07.228486 10.229.20.96 192.168.0.100 TLSv1.2 571

Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface eth0, Src: Vmware\_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco\_f6:1d:ae (00:be:75:f6:1d:ae) Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38 Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 517 Secure Sockets Layer

TLsv1.2 Record Layer: Handshake Protocol: Client Hello  
Content Type: Handshake (22)  
Version: TLS 1.0 (0x0301)  
Length: 512



72 2019-10-23 07:45:14.466836 192.168.0.100 72.163.4.38 TCP 74 35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK\_PERM=1 TSval=16127801 TSecr=0 WS=128

73 2019-10-23 07:45:14.632855 72.163.4.38 192.168.0.100 TCP 60 443 → 35752 [SYN, ACK] Seq=2770078884 Ack=2427943532 Win=8190 Len=0 MSS=1330

74 2019-10-23 07:45:14.632935 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [ACK] Seq=2427943532 Ack=2770078885 Win=29200 Len=0

75 2019-10-23 07:45:14.634091 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.com Client Hello

76 2019-10-23 07:45:14.634796 72.163.4.38 192.168.0.100 TCP 60 443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=32768 Len=0

77 2019-10-23 07:45:14.966729 72.163.4.38 192.168.0.100 TLSv1.2 150 Server Hello

78 2019-10-23 07:45:14.966722 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=29200 Len=0

79 2019-10-23 07:45:14.966834 72.163.4.38 192.168.0.100 TCP 1384 443 → 35752 [PSH, ACK] Seq=2427944049 Ack=2427944049 Win=32768 Len=1330 [TCP segment of a reassembled PDU]

80 2019-10-23 07:45:14.966850 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=31920 Len=0

81 2019-10-23 07:45:14.966877 72.163.4.38 192.168.0.100 TLSv1.2 155 Certificate

82 2019-10-23 07:45:14.966877 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [ACK] Seq=2427944049 Ack=2770080421 Win=31920 Len=0

83 2019-10-23 07:45:14.966915 72.163.4.38 192.168.0.100 TLSv1.2 63 Server Hello Done

84 2019-10-23 07:45:14.966925 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [ACK] Seq=2427944049 Ack=2770080421 Win=31920 Len=0

85 2019-10-23 07:45:14.967114 192.168.0.100 72.163.4.38 TLSv1.2 61 Alert (Level: Fatal, Description: Unknown CA)

86 2019-10-23 07:45:14.967261 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [RST, ACK] Seq=2427944056 Ack=2770080421 Win=31920 Len=0

87 2019-10-23 07:45:14.967382 72.163.4.38 192.168.0.100 TCP 60 443 → 35752 [ACK] Seq=2770080421 Ack=2427944056 Win=32768 Len=0

88 2019-10-23 07:45:14.967398 192.168.0.100 72.163.4.38 TCP 54 35752 → 443 [RST] Seq=2427944056 Win=0 Len=0

Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface eth0, Src: Vmware\_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco\_f6:1d:ae (00:be:75:f6:1d:ae) Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38 Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 517 Secure Sockets Layer

TLsv1.2 Record Layer: Handshake Protocol: Client Hello  
Content Type: Handshake (22)  
Version: TLS 1.0 (0x0301)  
Length: 512

Handshake Protocol: Client Hello  
Handshake Type: Client Hello (1)  
Length: 508  
Version: TLS 1.2 (0x0303)  
Random: 234490a107438c73b55564653271c7c09fbb7ac16897184...  
Session ID Length: 0  
Cipher Suites Length: 100  
Cipher Suites (50 suites)

## 요점:

1. TCP 3-way 핸드셰이크가 있습니다.
2. 클라이언트(FMC)는 Smart Licensing 포털에 SSL Client Hello 메시지를 보냅니다.
3. SSL 세션 ID는 0입니다. 다시 시작된 세션이 아님을 의미합니다.
4. 대상 서버가 Server Hello, Certificate 및 Server Hello Done 메시지로 응답합니다.
5. 클라이언트는 'Unknown CA'와 관련된 SSL Fatal Alert를 전송합니다.
6. 클라이언트는 세션을 닫기 위해 TCP RST를 전송합니다.
7. 전체 TCP 세션 기간(설정부터 종료까지)은 ~0.5초입니다.

Server Certificate(서버 인증서)를 선택하고 issuer(발급자) 필드를 확장하여 commonName을 확인합니다. 이 경우 Common Name(공통 이름)은 MITM(Man-in-the-middle)을 수행하는 디바이스를 나타냅니다.

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74		35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16127801
73	2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [SYN, ACK] Seq=2770078884 Ack=2427943532 Win=8190 Len=0 MSS=1330
74	2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427943532 Ack=2770078885 Win=29200 Len=0
75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
76	2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=32768 Len=0
77	2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150		Server Hello
78	2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=29200 Len=0
79	2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1384		443 → 35752 [PSH, ACK] Seq=2770078981 Ack=2427944049 Win=32768 Len=1330 [TCP segment
80	2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=31920 Len=0
81	2019-10-23 07:45:14.966872	72.163.4.38	192.168.0.100	TLSv1.2	155		Certificate

```

Length: 1426
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
      Length: 1422
        Certificates Length: 1419
          Certificates (1419 bytes)
            Certificate Length: 1416
              Certificate: 308205843082046ca003020102020d00aa23af5d607e0000... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose,id-at-sto
                signedCertificate
                  version: v3 (2)
                  serialNumber: 0x00aa23af5d607e00002f423880
                  > signature (sha256WithRSAEncryption)
                    > issuer: rdnSequence (0)
                      > rdnSequence: 3 items (id-at-commonName=FTD4100_MITM,id-at-organizationalUnitName=FTD_OU,id-at-organizationName=FTD_O)
                        > RDNSequene item: 1 item (id-at-organizationName=FTD_O)
                        > RDNSequene item: 1 item (id-at-organizationalUnitName=FTD_OU)
                        > RDNSequene item: 1 item (id-at-commonName=FTD4100_MITM)
                  > validity
                  > subject: rdnSequence (0)
                  > subjectPublicKeyInfo
                > extensions: 6 items
  
```

이 그림에서는 다음과 같이 표시됩니다.

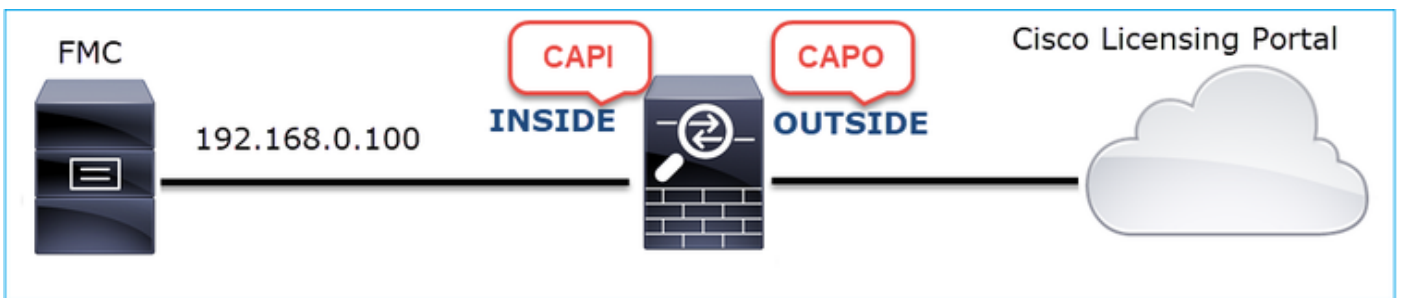


## 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 줄이기 위한 목적입니다.

### 작업 1. 추가 캡처 수행

트랜지트 방화벽 디바이스에서 캡처 수행:



CAPI의 특징은 다음과 같습니다.

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1221	2019-10-22 17:49:03.212681	192.168.0.100	173.37.145.8	TCP	74		39924 → 443 [SYN] Seq=427175838 Win=29200 Len=0 MSS=1460 SACK_PERM=1
1222	2019-10-22 17:49:03.379023	173.37.145.8	192.168.0.100	TCP	58		443 → 39924 [SYN, ACK] Seq=236460465 Ack=427175839 Win=8190 Len=0 MSS=1330
1223	2019-10-22 17:49:03.379298	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427175839 Ack=236460466 Win=29200 Len=0
1224	2019-10-22 17:49:03.380336	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
1225	2019-10-22 17:49:03.380732	173.37.145.8	192.168.0.100	TCP	54		443 → 39924 [ACK] Seq=236460466 Ack=427176356 Win=32768 Len=0
1226	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TLSv1.2	150		Server Hello
1227	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TCP	1384		443 → 39924 [PSH, ACK] Seq=236460562 Ack=427176356 Win=32768 Len=1330
1228	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TLSv1.2	155		Certificate
1229	2019-10-22 17:49:03.710107	173.37.145.8	192.168.0.100	TLSv1.2	63		Server Hello Done
1230	2019-10-22 17:49:03.710412	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236460562 Win=29200 Len=0
1231	2019-10-22 17:49:03.710519	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236461892 Win=31920 Len=0
1232	2019-10-22 17:49:03.710519	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236461993 Win=31920 Len=0
1233	2019-10-22 17:49:03.710534	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236462002 Win=31920 Len=0
1234	2019-10-22 17:49:03.710626	192.168.0.100	173.37.145.8	TLSv1.2	61		Alert (Level: Fatal, Description: Unknown CA)
1235	2019-10-22 17:49:03.710641	173.37.145.8	192.168.0.100	TCP	54		443 → 39924 [ACK] Seq=236462002 Ack=427176363 Win=32768 Len=0
1236	2019-10-22 17:49:03.710748	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [RST, ACK] Seq=427176363 Ack=236462002 Win=31920 Len=0
1237	2019-10-22 17:49:03.710870	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [RST] Seq=427176363 Win=0 Len=0

```

Length: 1426
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1422
    Certificates Length: 1419
  Certificates (1419 bytes)
    Certificate Length: 1416
  Certificate: 308205843082046ca003020102020d00aa23af5d607e0000... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose)
    signedCertificate
      version: v3 (2)
      serialNumber: 0x00aa23af5d607e00002f423880
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 3 items (id-at-commonName=FTD4100_MITM,id-at-organizationalUnitName=FTD_OU,id-at-organizationName=FTD_O)
          RDNSSequence item: 1 item (id-at-organizationName=FTD_O)
          RDNSSequence item: 1 item (id-at-organizationalUnitName=FTD_OU)
          RDNSSequence item: 1 item (id-at-commonName=FTD4100_MITM)
      validity
  
```

CAPO는 다음을 보여줍니다.

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1169	2019-10-22 17:49:03.212849	192.168.0.100	173.37.145.8	TCP	78		39924 → 443 [SYN] Seq=623942018 Win=29200 Len=0 MSS=1380 SACK_PERM=1 TSval=1169
1170	2019-10-22 17:49:03.378962	173.37.145.8	192.168.0.100	TCP	62		443 → 39924 [SYN, ACK] Seq=4179450724 Ack=623942019 Min=8190 Len=0 MSS=1330
1171	2019-10-22 17:49:03.379329	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942019 Ack=4179450725 Win=29200 Len=0
1172	2019-10-22 17:49:03.380793	192.168.0.100	173.37.145.8	TLSv1.2	512	tools.cisco.com	Client Hello
1173	2019-10-22 17:49:03.545748	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179450725 Ack=623942473 Win=34780 Len=1330 [TCP
1174	2019-10-22 17:49:03.545809	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179452055 Ack=623942473 Win=34780 Len=1330 [TCP
1175	2019-10-22 17:49:03.545824	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942473 Ack=4179453385 Win=65535 Len=0
1176	2019-10-22 17:49:03.545915	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179453385 Ack=623942473 Win=34780 Len=1330 [TCP
1177	2019-10-22 17:49:03.545961	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179454715 Ack=623942473 Win=34780 Len=1330 [TCP
1178	2019-10-22 17:49:03.545961	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942473 Ack=4179456045 Win=65535 Len=0
1179	2019-10-22 17:49:03.709420	173.37.145.8	192.168.0.100	TLSv1.2	82		Server Hello, Certificate, Server Hello Done
1180	2019-10-22 17:49:03.710687	192.168.0.100	173.37.145.8	TLSv1.2	65		Alert (Level: Fatal, Description: Unknown CA)
1181	2019-10-22 17:49:03.710885	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [FIN, PSH, ACK] Seq=623942480 Ack=4179456069 Win=65535 Len=0
1182	2019-10-22 17:49:03.874542	173.37.145.8	192.168.0.100	TCP	58		443 → 39924 [RST, ACK] Seq=4179456069 Ack=623942480 Win=9952 Len=0

```

Length: 5339
  Handshake Protocol: Server Hello
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 5240
    Certificates Length: 5237
  Certificates (5237 bytes)
    Certificate Length: 2025
  Certificate: 308207e5308205cda00302010202143000683b0f7504f7b2... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose)
    signedCertificate
      algorithmIdentifier (sha256WithRSAEncryption)
      padding: 0
      encrypted: 6921d084f7a6f6167058f14e2aad8b98b4e6c971ea6ea3b4...
    Certificate Length: 1736
  Certificate: 308206c4308204aca00302010202147517167783d0437eb5... (id-at-commonName=HydrantID SSL ICA G2,id-at-organizationName=HydrantID (Avalanche Cloud Corporation),id-at-localityName=San Jose)
    signedCertificate
      version: v3 (2)
      serialNumber: 0x7517167783d0437eb556c357946e4563b8ebd3ac
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 3 items (id-at-commonName=QuoVadis Root CA 2,id-at-organizationName=QuoVadis Limited,id-at-countryName=BM)
      validity
  
```

이러한 캡처는 트랜짓 방화벽이 MITM(서버 인증서)을 수정한다는 것을 입증합니다

작업 2. 디바이스 로그를 확인합니다.

이 문서에 설명된 대로 FMC TS 번들을 수집할 수 있습니다.

<https://www.cisco.com/c/en/us/support/docs/security/sourcefire-defense-center/117663-technote-SourceFire-00.html>

이 경우 /dir-archives/var-log/process\_stdout.log 파일에는 다음과 같은 메시지가 표시됩니다.

<#root>

```
SOUT: 10-23 05:45:14 2019-10-23 05:45:36 sla[10068]: *Wed .967 UTC: CH-LIB-ERROR: ch_pf_curl_send_msg[4]
failed to perform, err code 60, err string "SSL peer certificate or SSH remote key was not OK"
```

```
...
SOUT: 10-23 05:45:14 2019-10-23 05:45:36 sla[10068]: *Wed .967 UTC: CH-LIB-TRACE: ch_pf_curl_is_cert_is
cert issue checking, ret 60, url "https://tools.cisco.com/its/
```

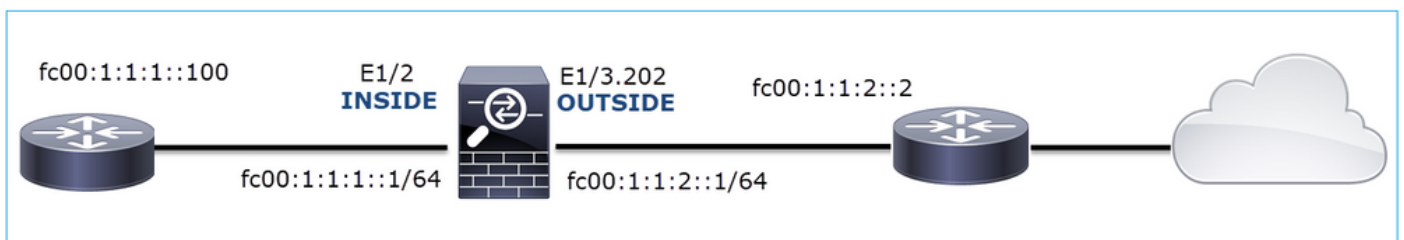
### 권장 솔루션

FMC가 Smart Licensing 클라우드에 성공적으로 등록할 수 있도록 특정 흐름에 대해 MITM을 비활성화합니다.

### 사례 11. IPv6 연결 문제

문제 설명: 방화벽의 INSIDE 인터페이스 뒤에 있는 내부 호스트는 외부 호스트(방화벽의 OUTSIDE 인터페이스 뒤에 있는 호스트)와 통신할 수 없습니다.

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: fc00:1:1:1::100

Dst IP: fc00:1:1:2::2

프로토콜: 모두

캡처 분석

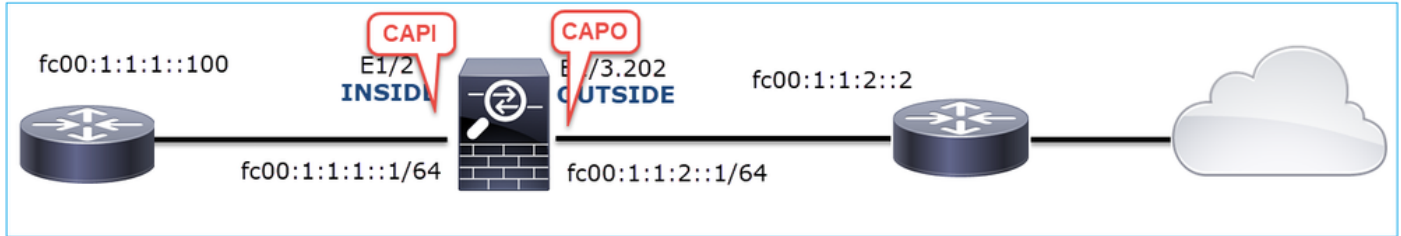
FTD LINA 엔진에서 캡처를 활성화합니다.



```

<#root>
firepower#
capture CAPI int INSIDE match ip any6 any6
firepower#
capture CAPO int OUTSIDE match ip any6 any6

```



캡처 - 작동하지 않는 시나리오

이러한 캡처는 IP fc00:1:1:1::100(내부 라우터)에서 IP fc00:1:1:2::2(업스트림 라우터)로의 ICMP 연결 테스트와 병행하여 수행되었습니다.

방화벽 INSIDE 인터페이스의 캡처에는 다음이 포함됩니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 13:02:07.001663	fc00:1:1:1::100	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
2	2019-10-24 13:02:07.001876	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6	86	Neighbor Advertisement fc00:1:1:1::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:ae
3	2019-10-24 13:02:07.002273	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=0, hop limit=64 (no response found!)
4	2019-10-24 13:02:08.997918	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
5	2019-10-24 13:02:10.998056	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
6	2019-10-24 13:02:11.999917	fe80::2be:75ff:fef6:1dae	fc00:1:1:1::100	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::100 from 00:be:75:f6:1d:ae
7	2019-10-24 13:02:12.002075	fc00:1:1:1::100	fe80::2be:75ff:fef6:1dae	ICMPv6	78	Neighbor Advertisement fc00:1:1:1::100 (rtr, sol)
8	2019-10-24 13:02:12.998346	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
9	2019-10-24 13:02:14.998483	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
10	2019-10-24 13:02:17.062725	fe80::4e4e:35ff:fe:fc:d8	fe80::2be:75ff:fef6:1dae	ICMPv6	86	Neighbor Solicitation for fe80::2be:75ff:fef6:1dae from 4c:4e:35:fc:fc:d8
11	2019-10-24 13:02:17.062862	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fe:fc:d8	ICMPv6	78	Neighbor Advertisement fe80::2be:75ff:fef6:1dae (rtr, sol)
12	2019-10-24 13:02:22.059994	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fe:fc:d8	ICMPv6	86	Neighbor Solicitation for fe80::4e4e:35ff:fe:fc:d8 from 00:be:75:f6:1d:ae
13	2019-10-24 13:02:22.063000	fe80::4e4e:35ff:fe:fc:d8	fe80::2be:75ff:fef6:1dae	ICMPv6	78	Neighbor Advertisement fe80::4e4e:35ff:fe:fc:d8 (rtr, sol)

요점:

1. 라우터가 IPv6 Neighbor Solicitation 메시지를 전송하고 업스트림 디바이스(IP fc00:1:1:1::1)의 MAC 주소를 요청합니다.
2. 방화벽이 IPv6 네이버 알림으로 응답합니다.
3. 라우터가 ICMP 에코 요청을 보냅니다.
4. 방화벽은 IPv6 인접 디바이스 요청 메시지를 전송하고 다운스트림 디바이스의 MAC 주소를 요청합니다(fc00:1:1:1::100).
5. 라우터가 IPv6 네이버 광고로 응답합니다.
6. 라우터가 추가 IPv6 ICMP 에코 요청을 보냅니다.

방화벽 OUTSIDE 인터페이스의 캡처에는 다음이 포함됩니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 13:02:07.002517	fe80::2be:75ff:fef6:1d8e	ff02::1:ff00:2	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
2	2019-10-24 13:02:07.005569	fc00:1:1:2::2	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
3	2019-10-24 13:02:08.997995	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	18	Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
4	2019-10-24 13:02:09.001815	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
5	2019-10-24 13:02:10.025938	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
6	2019-10-24 13:02:10.998132	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
7	2019-10-24 13:02:11.050015	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
8	2019-10-24 13:02:12.066082	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Solicitation for fe80::2be:75ff:fef6:1d8e from 4c:4e:35:fc:fc:d8
9	2019-10-24 13:02:12.066234	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	82	Neighbor Advertisement fe80::2be:75ff:fef6:1d8e (rtr, sol)
10	2019-10-24 13:02:12.998422	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
11	2019-10-24 13:02:13.002105	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
12	2019-10-24 13:02:14.090251	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
13	2019-10-24 13:02:14.998544	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
14	2019-10-24 13:02:15.178350	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
15	2019-10-24 13:02:17.059963	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	90	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8 from 00:be:75:f6:1d:8e
16	2019-10-24 13:02:17.062512	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	82	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

## 요점:

1. 방화벽은 업스트림 디바이스(IP fc00:1:1:2::2)의 MAC 주소를 묻는 IPv6 인접 디바이스 요청 메시지를 전송합니다.
2. 라우터가 IPv6 네이버 광고로 응답합니다.
3. 방화벽에서 IPv6 ICMP 에코 요청을 보냅니다.
4. 업스트림 디바이스(라우터 fc00:1:1:2::2)는 IPv6 주소 fc00:1:1:1::100의 MAC 주소를 묻는 IPv6 Neighbor Solicitation 메시지를 보냅니다.
5. 방화벽에서 추가 IPv6 ICMP 에코 요청을 보냅니다.
6. 업스트림 라우터는 IPv6 주소 fc00:1:1:1::100의 MAC 주소를 묻는 추가 IPv6 Neighbor Solicitation 메시지를 보냅니다.

포인트 4는 매우 흥미롭습니다. 일반적으로 업스트림 라우터는 방화벽 OUTSIDE 인터페이스(fc00:1:1:2::2)의 MAC을 요청하지만, 대신 fc00:1:1:1::100을 요청합니다. 이는 컨피그레이션이 잘못되었음을 나타냅니다.

## 권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. IPv6 인접 디바이스 테이블을 확인합니다.

방화벽 IPv6 인접 디바이스 테이블이 제대로 채워집니다.

```
<#root>
```

```
firepower#
```

```
show ipv6 neighbor | i fc00
```

```
fc00:1:1:2::2          58 4c4e.35fc.fcd8 STALE OUTSIDE
fc00:1:1:1::100       58 4c4e.35fc.fcd8 STALE INSIDE
```

작업 2. IPv6 컨피그레이션을 확인합니다.

이는 방화벽 컨피그레이션입니다.

```
<#root>
```

```
firewall#  
  
show run int e1/2  
  
!  
interface Ethernet1/2  
 nameif INSIDE  
 cts manual  
  propagate sgt preserve-untag  
  policy static sgt disabled trusted  
 security-level 0  
 ip address 192.168.0.1 255.255.255.0  
 ipv6 address  
  
fc00:1:1:1::1/64  
  
 ipv6 enable  
  
firewall#  
  
show run int e1/3.202  
  
!  
interface Ethernet1/3.202  
 vlan 202  
 nameif OUTSIDE  
 cts manual  
  propagate sgt preserve-untag  
  policy static sgt disabled trusted  
 security-level 0  
 ip address 192.168.103.96 255.255.255.0  
 ipv6 address  
  
fc00:1:1:2::1/64  
  
 ipv6 enable
```

업스트림 디바이스 컨피그레이션에서 잘못된 컨피그레이션을 확인합니다.

<#root>

```
Router#  
  
show run interface g0/0.202  
  
!  
interface GigabitEthernet0/0.202  
 encapsulation dot1Q 202  
 vrf forwarding VRF202  
 ip address 192.168.2.72 255.255.255.0  
 ipv6 address FC00:1:1:2::2  
  
/48
```

캡처 - 기능 시나리오

서브넷 마스크 변경(예: /48에서 /64로)으로 문제가 해결되었습니다. 기능 시나리오의 CAPI 캡처입

니다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 15:17:20.677775	fc00:1:1:1::100	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
2	2019-10-24 15:17:20.677989	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6	86	Neighbor Advertisement fc00:1:1:1::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:ae
3	2019-10-24 15:17:20.678401	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=0, hop limit=64 (no response found!)
4	2019-10-24 15:17:22.674281	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=1, hop limit=64 (no response found!)
5	2019-10-24 15:17:22.674403	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=2, hop limit=64 (reply in 6)
6	2019-10-24 15:17:24.674815	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	114	Echo (ping) reply id=0x097e, seq=2, hop limit=64 (request in 5)
7	2019-10-24 15:17:24.675242	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=3, hop limit=64 (reply in 8)
8	2019-10-24 15:17:24.675731	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	114	Echo (ping) reply id=0x097e, seq=3, hop limit=64 (request in 7)
9	2019-10-24 15:17:24.676356	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=4, hop limit=64 (reply in 10)
10	2019-10-24 15:17:24.676753	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	114	Echo (ping) reply id=0x097e, seq=4, hop limit=64 (request in 9)

요점:

1. 라우터가 업스트림 디바이스(IP fc00:1:1:1::1)의 MAC 주소를 묻는 IPv6 Neighbor Solicitation 메시지를 보냅니다.
2. 방화벽이 IPv6 네이버 알림으로 응답합니다.
3. 라우터가 ICMP 에코 요청을 보내고 에코 응답을 받습니다.

CAPO 내용:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 15:17:20.678645	fe80::2be:75ff:fe...	ff02::1:ff00:2	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
2	2019-10-24 15:17:20.681818	fc00:1:1:2::2	fe80::2be:75ff:fe...	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
3	2019-10-24 15:17:22.674342	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=1, hop limit=64 (reply in 6)
4	2019-10-24 15:17:22.677943	fc00:1:1:2::2	ff02::1:ff00:1	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::1 from 4c:4e:35:fc:fc:d8
5	2019-10-24 15:17:22.678096	fc00:1:1:2::1	fc00:1:1:2::2	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:8e
6	2019-10-24 15:17:22.678462	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=1, hop limit=64 (request in 3)
7	2019-10-24 15:17:24.674449	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=2, hop limit=64 (reply in 8)
8	2019-10-24 15:17:24.674785	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=2, hop limit=64 (request in 7)
9	2019-10-24 15:17:24.675395	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=3, hop limit=64 (reply in 10)
10	2019-10-24 15:17:24.675700	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=3, hop limit=64 (request in 9)
11	2019-10-24 15:17:24.676448	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=4, hop limit=64 (reply in 12)
12	2019-10-24 15:17:24.676738	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=4, hop limit=64 (request in 11)

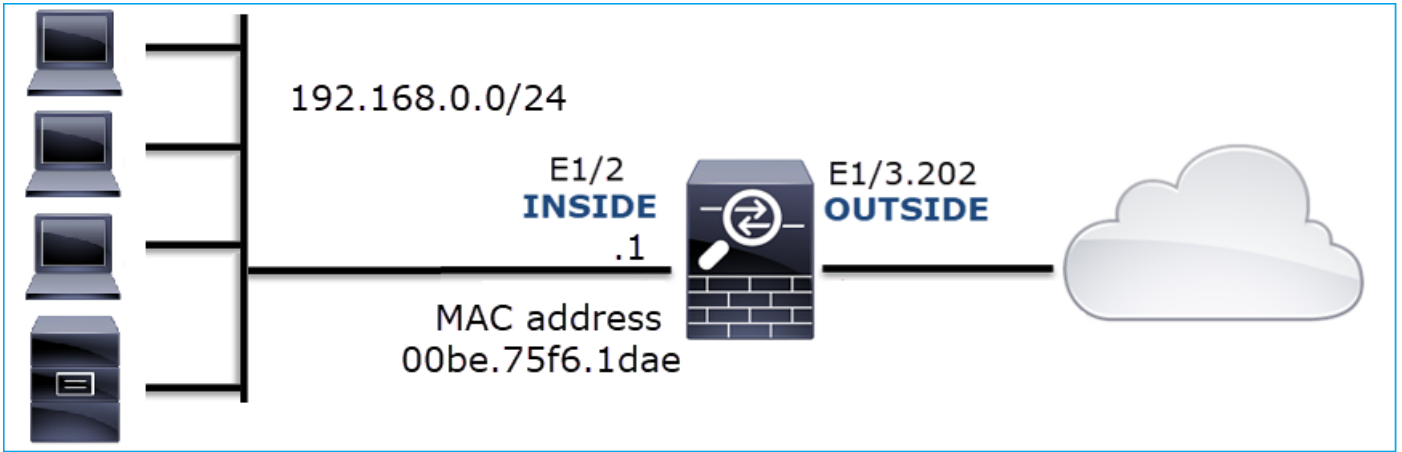
요점:

1. 방화벽은 업스트림 디바이스(IP fc00:1:1:2::2)의 MAC 주소를 묻는 IPv6 인접 디바이스 요청 메시지를 전송합니다.
2. 방화벽이 IPv6 네이버 알림으로 응답합니다.
3. 방화벽에서 ICMP 에코 요청을 보냅니다.
4. 라우터가 다운스트림 디바이스의 MAC 주소를 요청하는 IPv6 Neighbor Solicitation 메시지를 보냅니다(IP fc00:1:1:1::1).
5. 방화벽이 IPv6 네이버 알림으로 응답합니다.
6. 방화벽은 ICMP 에코 요청을 전송하고 에코 응답을 받습니다.

## 사례 12. 간헐적 연결 문제(ARP 중독)

문제 설명: 내부 호스트(192.168.0.x/24)에 동일한 서브넷에 있는 호스트와 가끔 연결 문제가 있습니다

이 그림에서는 토폴로지를 보여줍니다.



영향을 받는 흐름:

소스 IP: 192.168.0.x/24

Dst IP: 192.168.0.x/24

프로토콜: 모두

내부 호스트의 ARP 캐시가 다음과 같이 오염된 것 같습니다.

```

C:\Windows\system32\cmd.exe
C:\Users\mzafeirol>arp -a

Interface: 192.168.0.55 --- 0xb
Internet Address      Physical Address      Type
192.168.0.1           00-be-75-f6-1d-ae    dynamic
192.168.0.22          00-be-75-f6-1d-ae    dynamic
192.168.0.23          00-be-75-f6-1d-ae    dynamic
192.168.0.24          00-be-75-f6-1d-ae    dynamic
192.168.0.25          00-be-75-f6-1d-ae    dynamic
192.168.0.26          00-be-75-f6-1d-ae    dynamic
192.168.0.27          00-be-75-f6-1d-ae    dynamic
192.168.0.28          00-be-75-f6-1d-ae    dynamic
192.168.0.29          00-be-75-f6-1d-ae    dynamic
192.168.0.30          00-be-75-f6-1d-ae    dynamic
192.168.0.88          00-be-75-f6-1d-ae    dynamic
192.168.0.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250     01-00-5e-7f-ff-fa    static

C:\Users\mzafeirol>

```

캡처 분석

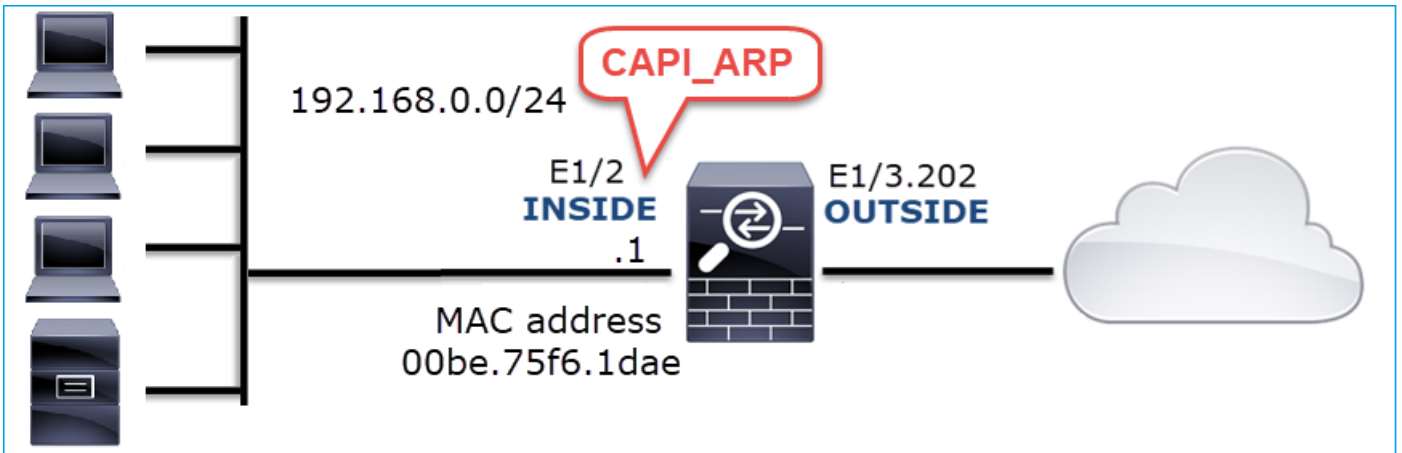
FTD LINA 엔진에서 캡처 활성화

이 캡처는 INSIDE 인터페이스의 ARP 패킷만 캡처합니다.

<#root>

firepower#

capture CAPI\_ARP interface INSIDE ethernet-type arp



캡처 - 비작동 시나리오:

방화벽 INSIDE 인터페이스의 캡처에는 다음이 포함됩니다.

No.	Time	Source	Destination	Protocol	Length	Info
4	2019-10-25 10:01:55.179571	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.23? Tell 192.168.0.55
5	2019-10-25 10:01:55.17969	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.23 is at 00:be:75:f6:1d:ae
35	2019-10-25 10:02:13.050397	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.24? Tell 192.168.0.55
36	2019-10-25 10:02:13.050488	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.24 is at 00:be:75:f6:1d:ae
47	2019-10-25 10:02:19.284683	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.25? Tell 192.168.0.55
48	2019-10-25 10:02:19.284775	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.25 is at 00:be:75:f6:1d:ae
61	2019-10-25 10:02:25.779821	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.26? Tell 192.168.0.55
62	2019-10-25 10:02:25.779912	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.26 is at 00:be:75:f6:1d:ae
76	2019-10-25 10:02:31.978175	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.27? Tell 192.168.0.55
77	2019-10-25 10:02:31.978251	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.27 is at 00:be:75:f6:1d:ae
97	2019-10-25 10:02:38.666515	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.28? Tell 192.168.0.55
98	2019-10-25 10:02:38.666606	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.28 is at 00:be:75:f6:1d:ae
121	2019-10-25 10:02:47.384074	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.29? Tell 192.168.0.55
122	2019-10-25 10:02:47.384150	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.29 is at 00:be:75:f6:1d:ae
137	2019-10-25 10:02:53.539995	Vmware_2c:9b:a7	Broadcast	ARP	60	Who has 192.168.0.30? Tell 192.168.0.55
138	2019-10-25 10:02:53.540087	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	42	192.168.0.30 is at 00:be:75:f6:1d:ae

요점:

1. 방화벽은 192.168.0.x/24 네트워크 내의 IP에 대한 다양한 ARP 요청을 수신합니다
2. 방화벽은 자신의 MAC 주소로 모든 응답(proxy-ARP)을 합니다

권장 작업

이 섹션에 나와 있는 조치는 문제를 더 줄이기 위한 목적입니다.

작업 1. NAT 컨피그레이션을 확인합니다.

NAT 컨피그레이션과 관련하여 no-proxy-arp 키워드가 이전 동작을 막을 수 있는 경우가 있습니다.

```
<#root>
```

```
firepower#
```

```
show run nat
```

```
nat (INSIDE,OUTSIDE) source static NET_1.1.1.0 NET_2.2.2.0 destination static NET_192.168.0.0 NET_4.4.4
```

no-proxy-arp

작업 2. 방화벽 인터페이스에서 proxy-arp 기능을 비활성화합니다.

'no-proxy-arp' 키워드로 문제가 해결되지 않으면 인터페이스 자체에서 프록시 ARP를 비활성화해 보십시오. FTD의 경우, 이 쓰기 시점에 FlexConfig를 사용하여 명령을 구축해야 합니다(적절한 인터페이스 이름 지정).

```
sysopt noproxyarp INSIDE
```

### 사례 13. CPU 호그를 유발하는 SNMP OID(Object Identifier) 식별

이 사례에서는 SNMP 버전 3(SNMPv3) 패킷 캡처 분석을 기반으로 메모리 폴링을 위한 특정 SNMP OID가 CPU 호그의 근본 원인(성능 문제)으로 어떻게 식별되었는지 보여줍니다.

문제 설명: 데이터 인터페이스에서 오버런이 계속 증가합니다. 추가 조사 결과, 인터페이스 오버런의 근본 원인인 CPU 호그(SNMP 프로세스로 인해 발생함)도 발견되었습니다.

트러블슈팅 프로세스의 다음 단계는 SNMP 프로세스로 인해 발생한 CPU 흡의 근본 원인을 파악하는 것이었습니다. 특히 문제의 범위를 좁혀 SNMP OID(Object Identifier)를 파악합니다. OID를 폴링하면 CPU 흡이 발생할 수 있습니다.

현재 FTD LINA 엔진은 실시간으로 폴링되는 SNMP OID에 대해 'show' 명령을 제공하지 않습니다.

폴링을 위한 SNMP OID 목록은 SNMP 모니터링 툴에서 검색할 수 있지만, 이 경우 다음과 같은 예방 요인이 있습니다.

- FTD 관리자가 SNMP 모니터링 툴에 액세스할 수 없었습니다
- 프라이버시를 위한 인증 및 데이터 암호화가 포함된 SNMP 버전 3이 FTD에서 구성되었습니다

#### 캡처 분석

FTD 관리자가 SNMP 버전 3 인증 및 데이터 암호화에 대한 자격 증명을 가지고 있으므로 다음 작업 계획을 제안했습니다.

1. SNMP 패킷 캡처
2. 캡처를 저장하고 Wireshark SNMP 프로토콜 기본 설정을 사용하여 SNMP 버전 3 자격 증명을 지정하여 SNMP 버전 3 패킷의 암호를 해독합니다. 해독된 캡처는 SNMP OID 분석 및 검색에 사용됩니다

snmp-server 호스트 컨피그레이션에 사용되는 인터페이스에서 SNMP 패킷 캡처를 구성합니다.

```
<#root>
```

```
firepower#
```

```
show run snmp-server | include host
```

```
snmp-server host management 192.168.10.10 version 3 netmonv3
```

```
firepower#
```

```
show ip address management
```

```
System IP Address:
```

Interface	Name	IP address	Subnet mask	Method
Management0/0	management	192.168.5.254	255.255.255.0	CONFIG

```
Current IP Address:
```

Interface	Name	IP address	Subnet mask	Method
Management0/0	management	192.168.5.254	255.255.255.0	CONFIG

```
firepower#
```

```
capture capsnpmp interface management buffer 10000000 match udp host 192.168.10.10 host 192.168.5.254 eq
```

```
firepower#
```

```
show capture capsnpmp
```

```
capture capsnpmp type raw-data buffer 10000000 interface outside [Capturing -
```

```
9512
```

```
bytes]
```

```
match udp host 192.168.10.10 host 192.168.5.254 eq snmp
```



No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	encryptedPDU: privKey Unknown
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	encryptedPDU: privKey Unknown
6	0.325	SNMP	192.168.5.254	161	65484	192.168.10.10	678	encryptedPDU: privKey Unknown
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	encryptedPDU: privKey Unknown
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	encryptedPDU: privKey Unknown
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	encryptedPDU: privKey Unknown
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	encryptedPDU: privKey Unknown
15	1.317	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
16	1.318	SNMP	192.168.5.254	161	65484	192.168.10.10	513	encryptedPDU: privKey Unknown
17	17.595	SNMP	192.168.10.10	62008	161	192.168.5.254	100	getBulkRequest
18	17.595	SNMP	192.168.5.254	161	62008	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
19	17.749	SNMP	192.168.10.10	62008	161	192.168.5.254	197	encryptedPDU: privKey Unknown
20	17.749	SNMP	192.168.5.254	161	62008	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
21	17.898	SNMP	192.168.10.10	62008	161	192.168.5.254	199	encryptedPDU: privKey Unknown
22	17.899	SNMP	192.168.5.254	161	62008	192.168.10.10	678	encryptedPDU: privKey Unknown
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	encryptedPDU: privKey Unknown
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	encryptedPDU: privKey Unknown
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	encryptedPDU: privKey Unknown

```

<[Destination Host: 192.168.5.254]>
<[Source or Destination Host: 192.168.5.254]>
> User Datagram Protocol, Src Port: 65484, Dst Port: 161
  Simple Network Management Protocol
    msgVersion: snmpv3 (3)
    > msgGlobalData
    > msgAuthoritativeEngineID: 80000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40
    msgAuthoritativeEngineBoots: 0
    msgAuthoritativeEngineTime: 0
    msgUserName: netmonv3
    msgAuthenticationParameters: ff5176f5973c30b62ffc11b8
    msgPrivacyParameters: 000040e100003196
    > msgData: encryptedPDU (1)
      encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703
  
```

요점:

1. SNMP 소스 및 목적지 주소/포트.
2. privKey가 Wireshark에 알려지지 않았으므로 SNMP 프로토콜 PDU를 디코딩할 수 없습니다.
3. encryptedPDU primitive의 값입니다.

권장 작업

이 섹션에 나와 있는 조치는 문제를 더 좁히기 위한 목적입니다.

작업 1. SNMP 캡처를 해독합니다.

를 저장하고 Wireshark SNMP 프로토콜 기본 설정을 편집하여 패킷 해독을 위한 SNMP 버전 3 자격 증명을 지정합니다.

```
</root>
```

```
firepower#
```

```
copy /pcap capture: tftp:
```

```
Source capture name [capsnmp]?
```

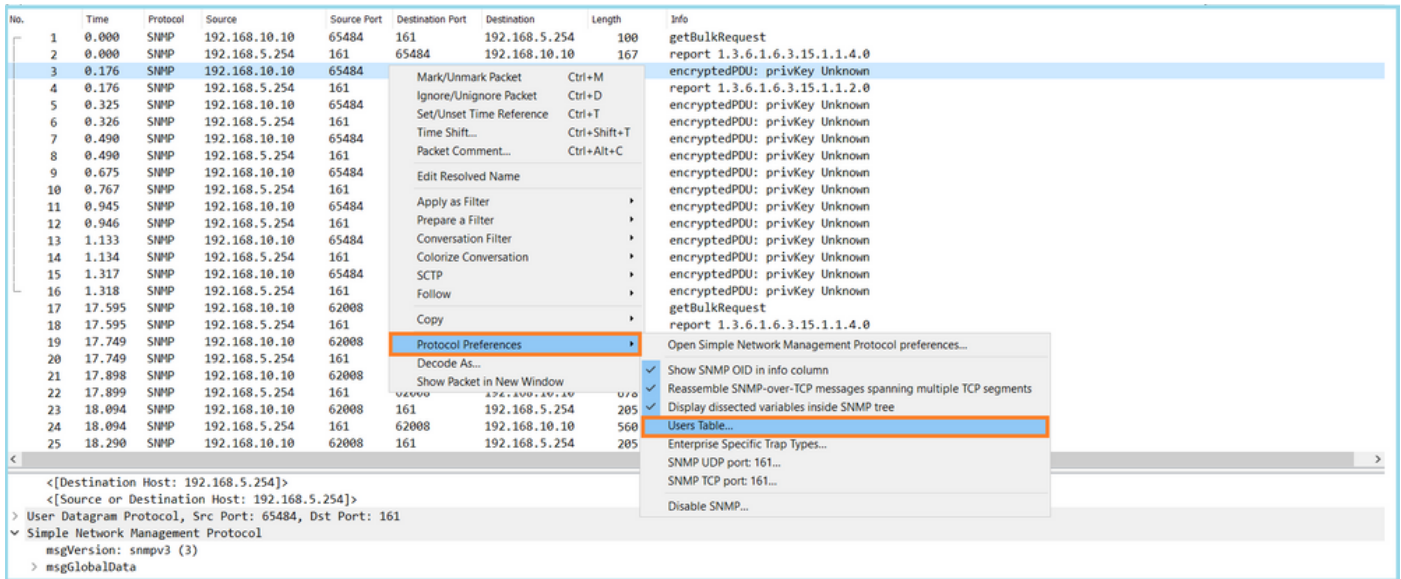
```
Address or name of remote host []? 192.168.10.253
```

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

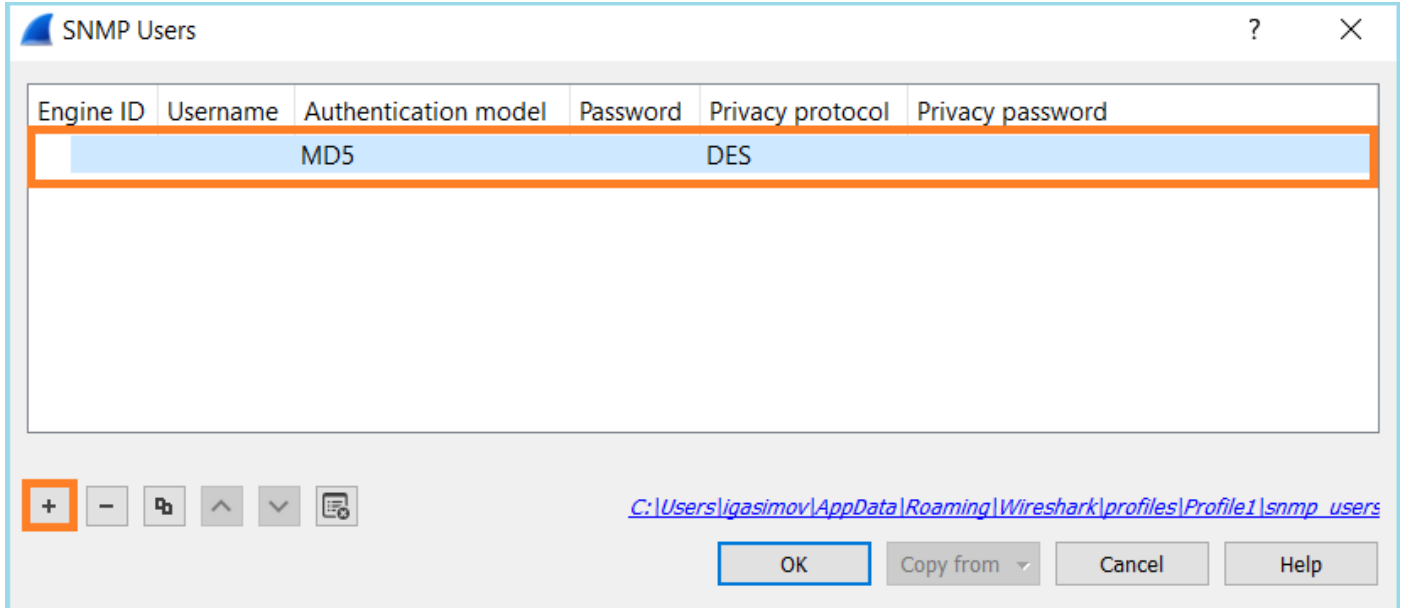
```
!!!!!!
```

```
64 packets copied in 0.40 secs
```

이미지에 표시된 대로 Wireshark에서 캡처 파일을 열고 SNMP 패킷을 선택하고 Protocol Preferences > Users Table로 이동합니다.



SNMP Users(SNMP 사용자) 테이블에 SNMP 버전 3 사용자 이름, 인증 모델, 인증 비밀번호, 프라이버시 프로토콜 및 프라이버시 비밀번호가 지정되었습니다(실제 자격 증명은 아래에 표시되지 않음).



SNMP 사용자 설정이 적용되면 Wireshark는 암호 해독된 SNMP PDU를 보여줍니다.

No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	getBulkRequest 1.3.6.1.4.1.9.9.221.1
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	getBulkRequest 1.3.6.1.4.1.9.9.221.1
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.7.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.8.1.8
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.8.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.17.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.17.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.18.1.8
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.18.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.19.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.19.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.20.1.8
15	1.317	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.20.1.8
16	1.318	SNMP	192.168.5.254	161	65484	192.168.10.10	513	get-response 1.3.6.1.4.1.9.9.392.1.1.1.0 1.3.6.1.4.1.9.9.392.1.1.2.0 1.3.6.1.4.1.9.9.392.1.1.3.0 1.3.6.1.4.1.9.9.392.1.1.4.0
17	17.595	SNMP	192.168.10.10	62008	161	192.168.5.254	100	getBulkRequest
18	17.595	SNMP	192.168.5.254	161	62008	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
19	17.749	SNMP	192.168.10.10	62008	161	192.168.5.254	197	getBulkRequest 1.3.6.1.4.1.9.9.221.1
20	17.749	SNMP	192.168.5.254	161	62008	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
21	17.898	SNMP	192.168.10.10	62008	161	192.168.5.254	199	getBulkRequest 1.3.6.1.4.1.9.9.221.1
22	17.899	SNMP	192.168.5.254	161	62008	192.168.10.10	678	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8

```

< msgData: encryptedPDU (1)
  encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703...
    Decrypted ScopedPDU: 303b04198000009fe1c6dad4930a00ef1fec2301621a415...
      contextEngineID: 8000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40...
      contextName:
      data: getBulkRequest (5)
        getBulkRequest
          request-id: 5620
          non-repeaters: 0
          max-repetitions: 16
        variable-bindings: 1 item
          1.3.6.1.4.1.9.9.221.1: Value (Null)
            Object Name: 1.3.6.1.4.1.9.9.221.1 (iso.3.6.1.4.1.9.9.221.1)
            Value (Null)
  
```

**요점:**

1. SNMP 모니터링 도구는 SNMP getBulkRequest를 사용하여 상위 OID 1.3.6.1.4.1.9.9.221.1 및 관련 OID를 쿼리하고 해당 OID를 탐색했습니다.
2. FTD는 1.3.6.1.4.1.9.9.221.1과 관련된 OID를 포함하는 get-response로 각 getBulkRequest에 응답했습니다.

**작업 2. SNMP OID를 식별합니다.**

[SNMP Object Navigator에서](#)는 다음과 같이 OID 1.3.6.1.4.1.9.9.221.1이 CISCO-ENHANCED-MEMPOOL-MIB라는 MIB(Management Information Base)에 속한다는 것을 보여 주었습니다.



Tools & Resources  
**SNMP Object Navigator**

HOME | TRANSLATE/BROWSE | SEARCH | **DOWNLOAD MIBS** | MIB SUPPORT - SW | Help | Feedback

Support | TOOLS & RESOURCES | **SNMP Object Navigator**

**CISCO-ENHANCED-MEMPOOL-MIB**

View compiling dependencies for other MIBS by [clearing](#) the page and selecting another MIB.

Compile the MIB

Before you can compile CISCO-ENHANCED-MEMPOOL-MIB, you need to compile the MIBs listed below in the order listed.

Download all of these MIBs (Warning: does not include non-Cisco MIBs) or view details about each MIB below.

If you are using Internet Explorer click [here](#).

MIB Name	Version 1	Version 2	Dependencies
1. SNMPv2-SMI	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">View Dependencies</a>
2. SNMPv2-TC	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">View Dependencies</a>
3. SNMPv2-CONF	Not Required	<a href="#">Download</a>	<a href="#">View Dependencies</a>
4. SNMP-FRAMEWORK-MIB	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">View Dependencies</a>
5. CISCO-SMI	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">View Dependencies</a>
6. ENTITY-MIB	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">View Dependencies</a>
7. HCNUM-TC	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">View Dependencies</a>
8. RFC1155-SMI	Non-Cisco MIB	Non-Cisco MIB	-
9. RFC-1212	Non-Cisco MIB	Non-Cisco MIB	-
10. RFC-1215	Non-Cisco MIB	Non-Cisco MIB	-
11. SNMPv2-TC-v1	Non-Cisco MIB	Non-Cisco MIB	-
12. CISCO-ENHANCED-MEMPOOL-MIB	<a href="#">Download</a>	<a href="#">Download</a>	

2. Wireshark의 편집 > 기본 설정 > 이름 확인 창에서 OID 확인 사용이 선택됩니다. SMI(MIB 및 PIB 경로) 창에서 다운로드된 MIB가 있는 폴더와 SMI(MIB 및 PIB 모듈)를 지정합니다. CISCO-ENHANCED-MEMPOOL-MIB는 모듈 목록에 자동으로 추가됩니다.

The screenshot shows the Wireshark interface with two configuration windows open:

- Name Resolution:** The 'Enable OID resolution' checkbox is checked and highlighted with an orange box.
- SMI Paths:** The 'Directory path' field is set to 'C:/Users/Administrator/Downloads/SNMPMIBS' and is highlighted with an orange box.
- SMI Modules:** The 'Module name' list includes 'CISCO-ENHANCED-MEMPOOL-MIB', which is highlighted with an orange box.

3. Wireshark를 다시 시작하면 OID 확인이 활성화됩니다.

No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report SNMP-USER-BASED-SM-MIB::usmStatsUnknownEngineIDs.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMIBObjects
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report SNMP-USER-BASED-SM-MIB::usmStatsNotInTimeWindows.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMIBObjects
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolType.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolType
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolAlternate.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoc
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolValid.1.8
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUsed.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUsed
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolFree.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUsedOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPc
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolHCUsed.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	600	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolFreeOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemP

```

CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.1 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.1): System memory
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.1 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.1)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: System memory
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.2 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.2): System memory
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.2 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.2)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: System memory
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.3 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.3): MEMPOOL_MSGLYR
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.3 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.3)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_MSGLYR
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.4 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.4): MEMPOOL_HEAPCACHE_1
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.4 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.4)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_HEAPCACHE_1
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.5 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.5): MEMPOOL_HEAPCACHE_0
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.5 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.5)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_HEAPCACHE_0
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.6 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.6): MEMPOOL_DMA_ALT1
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.6 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.6)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_DMA_ALT1
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.7 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.7): MEMPOOL_DMA
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.7 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.7)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_DMA
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.8 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.8): MEMPOOL_GLOBAL_SHARED
Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.8)
CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_GLOBAL_SHARED

```

캡처 파일의 해독된 출력을 기반으로 SNMP 모니터링 도구는 FTD의 메모리 풀 사용률에 대한 데이터를 정기적으로(10초 간격) 폴링했습니다. TechNote 문서 [ASA SNMP Polling for Memory-Related Statistics](#)([메모리 관련 통계에 대한 ASA SNMP 폴링](#))에서 설명한 것처럼, SNMP로 GSP(전역 공유 풀) 활용률을 폴링하면 CPU 사용률이 높아집니다. 이 경우 캡처를 통해 글로벌 공유 풀 사용률이 SNMP getBulkRequest primitive의 일부로 주기적으로 폴링되었음을 확인할 수 있습니다.

SNMP 프로세스로 인한 CPU 호그를 최소화하기 위해 기사에 언급된 SNMP용 CPU 호그에 대한 완화 단계를 따르고 GSP와 관련된 OID를 폴링하지 않는 것이 좋습니다. GSP와 관련된 OID에 대한 SNMP 폴링이 없으면 SNMP 프로세스로 인한 CPU 흡이 관찰되지 않았으며 오버런 비율이 크게 감소했습니다.

## 관련 정보

- [Cisco Firepower Management Center 컨피그레이션 가이드](#)
- [Firepower Threat Defense 액세스 제어 정책 규칙 작업 확인](#)
- [firepower Threat Defense 캡처 및 패킷 추적기 사용](#)
- [Wireshark 학습](#)

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.