FTD(Firepower Threat Defense) 캡처 및 패킷 트레이서 사용

목차

소개

이 문서에서는 FTD(Firepower Threat Defense) 캡처 및 패킷 트레이서 유틸리티를 사용하는 방법을 설명합니다.

패킷 캡처는 흔히 사용되는 트러블슈팅 툴 중 하나입니다. 패킷 캡처의 활용 사례는 다음과 같습니다.

- 패킷이 디바이스에 도착했음을 증명
- 패킷이 디바이스에서 나갔음을 증명
- 디바이스에서 패킷을 삭제했음을 증명(예: ASA ASP 삭제)

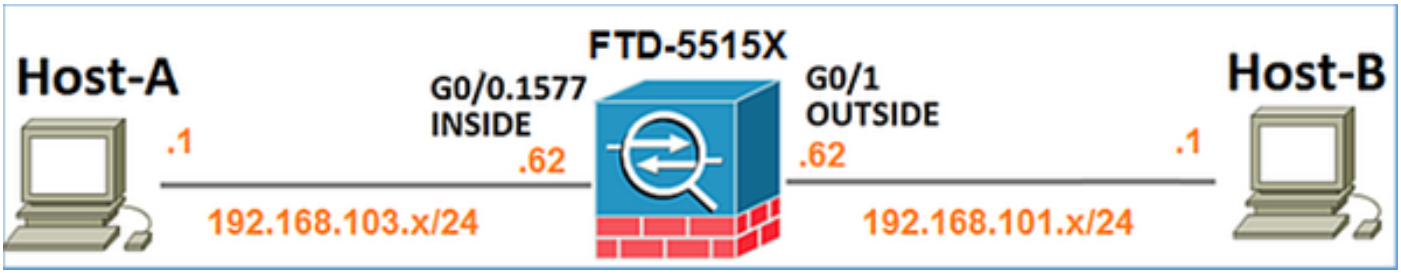
FTD에서는 다음의 두 엔진을 통해 패킷을 캡처할 수 있습니다.

1. ASA 엔진
2. Snort 엔진

사용되는 구성 요소

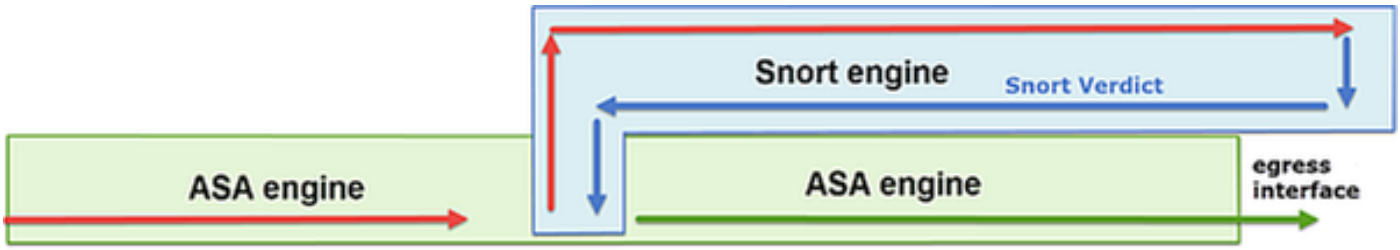
- FTD 코드 6.1.0(빌드 330)을 실행하는 ASA5515X
- 6.1.0(빌드 330)을 실행하는 FMC(Firepower Management Center)

토폴로지



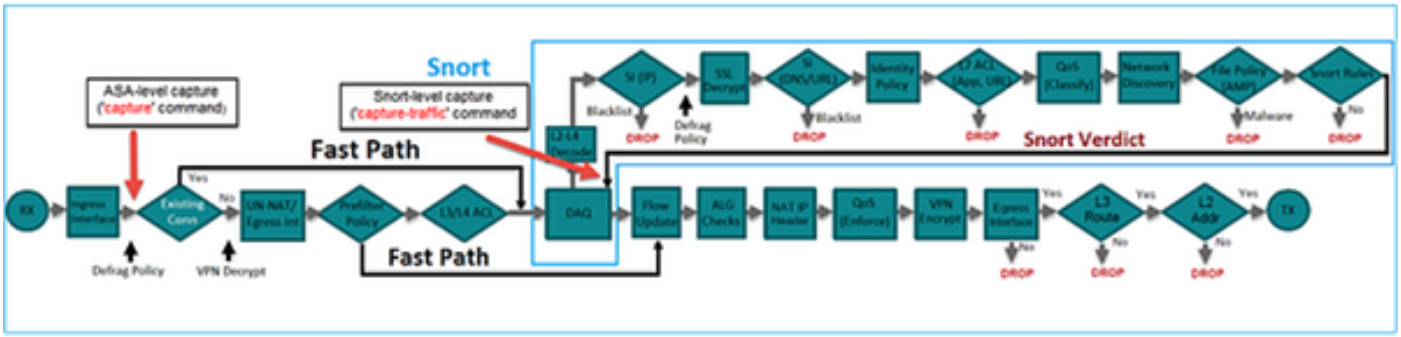
FTD 패킷 처리

FTD 패킷 처리는 다음 그림과 같이 표시할 수 있습니다.



1. 패킷이 인그레스 인터페이스로 들어와 ASA 엔진에 의해 처리됩니다.
2. 정책에서 지시하는 경우 Snort 엔진이 패킷을 검사합니다.
3. Snort 엔진이 패킷에 대해 화이트리스트, 블랙리스트 등의 판정을 반환합니다.
4. Snort의 판정을 기반으로 ASA 엔진이 패킷을 삭제하거나 전달합니다.

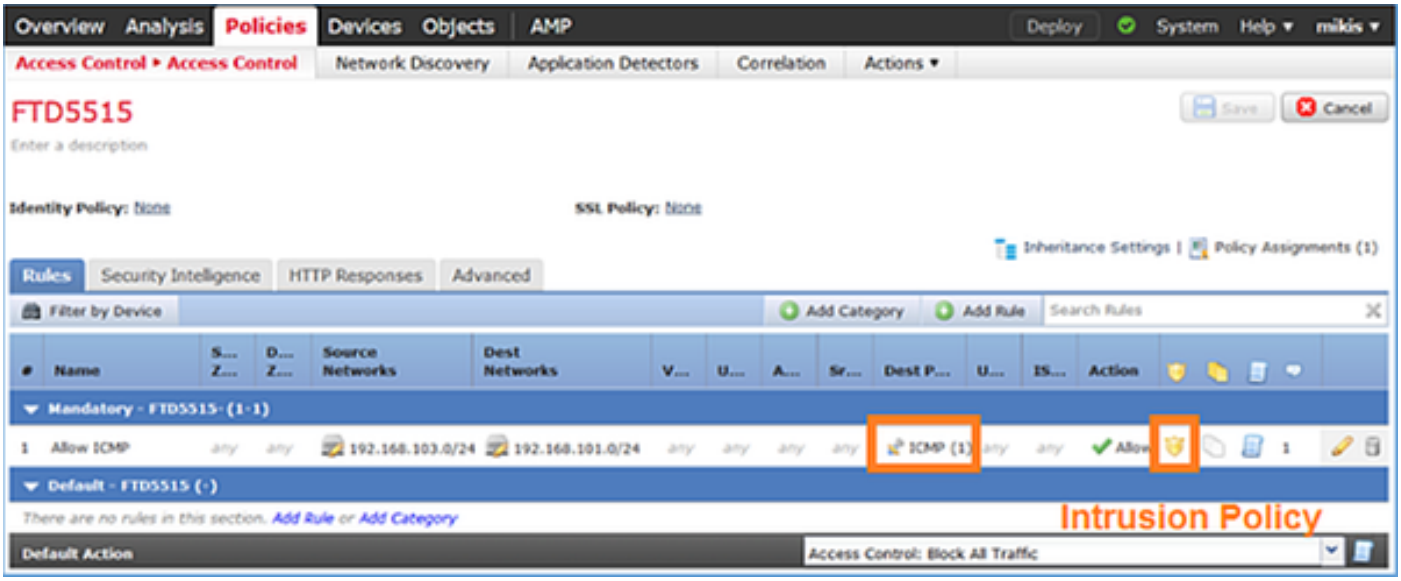
위의 아키텍처를 기준으로 할 때 서로 다른 두 위치에서 FTD 캡처를 가져올 수 있습니다.



Snort 엔진 캡처 사용

사전 요구 사항

FTD에서는 ICMP 트래픽 통과를 허용하는 ACP(액세스 제어 정책)가 적용됩니다. 이 정책에서는 침입 정책도 적용됩니다.



요건

1. 필터를 사용하지 않고 FTD CLISH 모드에서 캡처를 활성화합니다.
2. FTD에서 ping을 수행하여 캡처 출력을 확인합니다.

해결책

1단계: FTD 콘솔에 로그인하거나 SSH를 통해 br1 인터페이스에 액세스하여 필터를 사용하지 않고 FTD CLISH 모드에서 캡처를 활성화합니다.

```
> 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:
```

FTD 6.0.x에서 이 작업을 수행하는 명령은 다음과 같습니다.

```
> system support capture-traffic
```

2단계: FTD에서 ping을 수행하여 캡처 출력을 확인합니다.

```
> 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:
12:52:34.749945 IP olab-v1603-gw.cisco.com > olab-v1647-gw.cisco.com: ICMP echo request, id 0,
seq 1, length 80 12:52:34.749945 IP olab-v1647-gw.cisco.com > olab-v1603-gw.cisco.com: ICMP echo
reply, id 0, seq 1, length 80 12:52:34.759955 IP olab-v1603-gw.cisco.com > olab-v1647-
gw.cisco.com: ICMP echo request, id 0, seq 2, length 80 12:52:34.759955 IP olab-v1647-
gw.cisco.com > olab-v1603-gw.cisco.com: ICMP echo reply, id 0, seq 2, length 80 12:52:34.759955
IP olab-v1603-gw.cisco.com > olab-v1647-gw.cisco.com: ICMP echo request, id 0, seq 3, length 80
12:52:34.759955 IP olab-v1647-gw.cisco.com > olab-v1603-gw.cisco.com: ICMP echo reply, id 0, seq
3, length 80 12:52:34.759955 IP olab-v1603-gw.cisco.com > olab-v1647-gw.cisco.com: ICMP echo
request, id 0, seq 4, length 80 12:52:34.759955 IP olab-v1647-gw.cisco.com > olab-v1603-
gw.cisco.com: ICMP echo reply, id 0, seq 4, length 80 ^C <- to exit press CTRL + C
```

tcpdump 필터를 통해 Snort 엔진 캡처 사용

요건

1. IP 192.168.101.1에 대해 필터를 사용하여 FTD CLISH 모드에서 캡처를 활성화합니다.
2. FTD에서 ping을 수행하여 캡처 출력을 확인합니다.

해결책

1단계: IP 192.168.101.1에 대해 필터를 사용하여 FTD CLISH 모드에서 캡처를 활성화합니다.

```
> 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:
host 192.168.101.1
```

2단계: FTD에서 ping을 수행하여 캡처 출력을 확인합니다.

```
13:28:36.079982 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 3, seq
0, length 80
13:28:36.079982 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 3, seq
1, length 80
13:28:36.079982 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 3, seq
2, length 80
13:28:36.079982 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 3, seq
3, length 80
13:28:36.079982 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 3, seq
4, length 80
```

'-n' 옵션을 사용하면 호스트 및 포트 번호를 숫자 형식으로 확인할 수 있습니다. 예를 들어 위의 캡처는 다음과 같이 표시됩니다.

```
> 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 13:29:59.599959 IP 192.168.101.1 > 192.168.103.1: ICMP echo reply, id 5,
seq 0, length 80 13:29:59.599959 IP 192.168.101.1 > 192.168.103.1: ICMP echo reply, id 5, seq 1,
length 80 13:29:59.599959 IP 192.168.101.1 > 192.168.103.1: ICMP echo reply, id 5, seq 2, length
80 13:29:59.599959 IP 192.168.101.1 > 192.168.103.1: ICMP echo reply, id 5, seq 3, length 80
13:29:59.599959 IP 192.168.101.1 > 192.168.103.1: ICMP echo reply, id 5, seq 4, length 80
```

Tcpdump 필터 예

예 1

소스 IP 또는 목적지 IP = 192.168.101.1 및 소스 포트 또는 목적지 포트 = TCP/UDP 23을 캡처하려는 경우:

```
Options: -n host 192.168.101.1 and port 23
```

예 2

소스 IP = 192.168.101.1 및 소스 포트 = TCP/UDP 23을 캡처하려는 경우:

```
Options: -n src 192.168.101.1 and src port 23
```

예 3

소스 IP = 192.168.101.1 및 소스 포트 = TCP 23을 캡처하려는 경우:

```
Options: -n src 192.168.101.1 and tcp and src port 23
```

예 4

소스 IP = 192.168.101.1을 캡처하고 패킷의 MAC 주소를 확인하여 'e' 옵션을 추가하려는 경우:

```
Options: -ne src 192.168.101.1 17:57:48.709954 6c:41:6a:a1:2b:f6 > a8:9d:21:93:22:90, ethertype
IPv4 (0x0800), length 58: 192.168.101.1.23 > 192.168.103.1.25420: Flags [S.], seq 3694888749,
ack 1562083610, win 8192, options [mss 1380], length 0
```

예 5

패킷 10개를 캡처한 후 종료하려는 경우:

```
Options: -n -c 10 src 192.168.101.1 18:03:12.749945 IP 192.168.101.1.23 > 192.168.103.1.27287:
Flags [.], ack 3758037348, win 32768, length 0 18:03:12.749945 IP 192.168.101.1.23 >
192.168.103.1.27287: Flags [P.], ack 1, win 32768, length 2 18:03:12.949932 IP 192.168.101.1.23
> 192.168.103.1.27287: Flags [P.], ack 1, win 32768, length 10 18:03:13.249971 IP
192.168.101.1.23 > 192.168.103.1.27287: Flags [.], ack 3, win 32768, length 0 18:03:13.249971 IP
192.168.101.1.23 > 192.168.103.1.27287: Flags [P.], ack 3, win 32768, length 2 18:03:13.279969
IP 192.168.101.1.23 > 192.168.103.1.27287: Flags [.], ack 5, win 32768, length 0 18:03:13.279969
IP 192.168.101.1.23 > 192.168.103.1.27287: Flags [P.], ack 5, win 32768, length 10
18:03:13.309966 IP 192.168.101.1.23 > 192.168.103.1.27287: Flags [.], ack 7, win 32768, length 0
18:03:13.309966 IP 192.168.101.1.23 > 192.168.103.1.27287: Flags [P.], ack 7, win 32768, length
12 18:03:13.349972 IP 192.168.101.1.23 > 192.168.103.1.27287: Flags [.], ack 9, win 32768,
length 0
```

예 6

이름이 capture.pcap인 파일에 캡처를 쓰고 FTP를 통해 원격 서버에 복사하려는 경우:

```
Options: -w capture.pcap host 192.168.101.1 CTRL + C <- to stop the capture > system file copy
10.229.22.136 ftp / capture.pcap Enter password for ftp@10.229.22.136: Copying capture.pcap Copy
successful. >
```

FTD ASA 엔진 캡처 사용

요건

1. 다음 필터를 사용하여 FTD에서 캡처 2개를 활성화합니다.

Source IP(소스 IP)	192.168.103.1
Destination IP(목적지 IP)	192.168.101.1
프로토콜	ICMP
인터페이스	내부
Source IP(소스 IP)	192.168.103.1
Destination IP(목적지 IP)	192.168.101.1
프로토콜	ICMP
인터페이스	외부

2. 호스트-A(192.168.103.1) 및 호스트-B(192.168.101.1)에서 ping을 수행하고 캡처를 확인합니다.

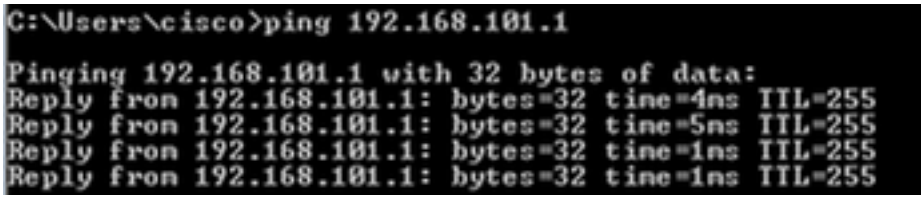
해결책

1단계: 캡처를 활성화합니다.

```
> capture CAPI interface INSIDE match icmp host 192.168.103.1 host 192.168.101.1 > capture CAPO interface OUTSIDE match icmp host 192.168.101.1 host 192.168.103.1
```

2단계: CLI를 사용하여 캡처를 확인합니다.

호스트-A에서 호스트-B로의 ping:




```
> show capture capture CAPI type raw-data interface INSIDE [Capturing - 752 bytes] match icmp host 192.168.103.1 host 192.168.101.1 capture CAPO type raw-data interface OUTSIDE [Capturing - 720 bytes] match icmp host 192.168.101.1 host 192.168.103.1
```

내부 인터페이스의 Dot1Q 헤더로 인해 2개 캡처는 크기가 서로 다릅니다. 다음 출력에서 이러한 크기의 차이를 표시할 수 있습니다.

```
> show capture CAPI 8 packets captured 1: 17:24:09.122338 802.1Q vlan#1577 P0 192.168.103.1 > 192.168.101.1: icmp: echo request 2: 17:24:09.123071 802.1Q vlan#1577 P0 192.168.101.1 > 192.168.103.1: icmp: echo reply 3: 17:24:10.121392 802.1Q vlan#1577 P0 192.168.103.1 > 192.168.101.1: icmp: echo request 4: 17:24:10.122018 802.1Q vlan#1577 P0 192.168.101.1 > 192.168.103.1: icmp: echo reply 5: 17:24:11.119714 802.1Q vlan#1577 P0 192.168.103.1 > 192.168.101.1: icmp: echo request 6: 17:24:11.120324 802.1Q vlan#1577 P0 192.168.101.1 > 192.168.103.1: icmp: echo reply 7: 17:24:12.133660 802.1Q vlan#1577 P0 192.168.103.1 > 192.168.101.1: icmp: echo request 8: 17:24:12.134239 802.1Q vlan#1577 P0 192.168.101.1 > 192.168.103.1: icmp: echo reply 8 packets shown > show capture CAPO 8 packets captured 1: 17:24:09.122765 192.168.103.1 > 192.168.101.1: icmp: echo request 2: 17:24:09.122994 192.168.101.1 > 192.168.103.1: icmp: echo reply 3: 17:24:10.121728 192.168.103.1 > 192.168.101.1: icmp: echo request 4: 17:24:10.121957 192.168.101.1 > 192.168.103.1: icmp: echo reply 5: 17:24:11.120034 192.168.103.1 > 192.168.101.1: icmp: echo request 6: 17:24:11.120263 192.168.101.1 > 192.168.103.1: icmp: echo reply 7: 17:24:12.133980 192.168.103.1 > 192.168.101.1: icmp: echo request 8: 17:24:12.134194 192.168.101.1 > 192.168.103.1: icmp: echo reply 8 packets shown
```

FTD ASA 엔진 캡처 사용 - HTTP를 사용하여 캡처 내보내기

요건

브라우저를 사용하여 이전 시나리오에서 가져온 캡처를 내보냅니다.

해결책

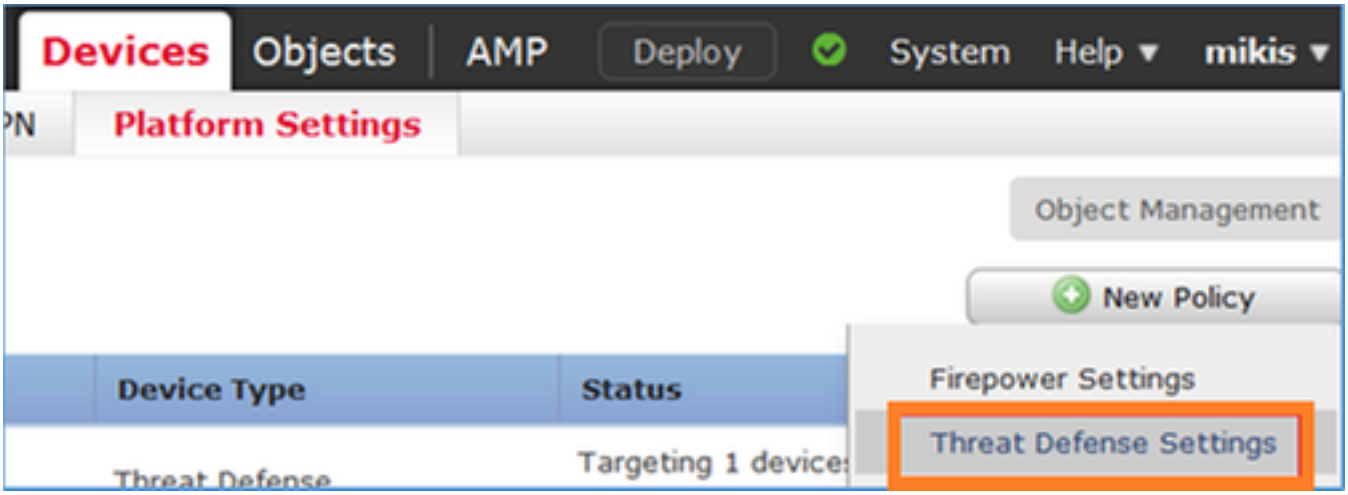
브라우저를 사용하여 캡처를 내보내려면 다음 작업을 수행해야 합니다.

1. HTTPS 서버를 활성화합니다.
2. HTTP 액세스를 허용합니다.

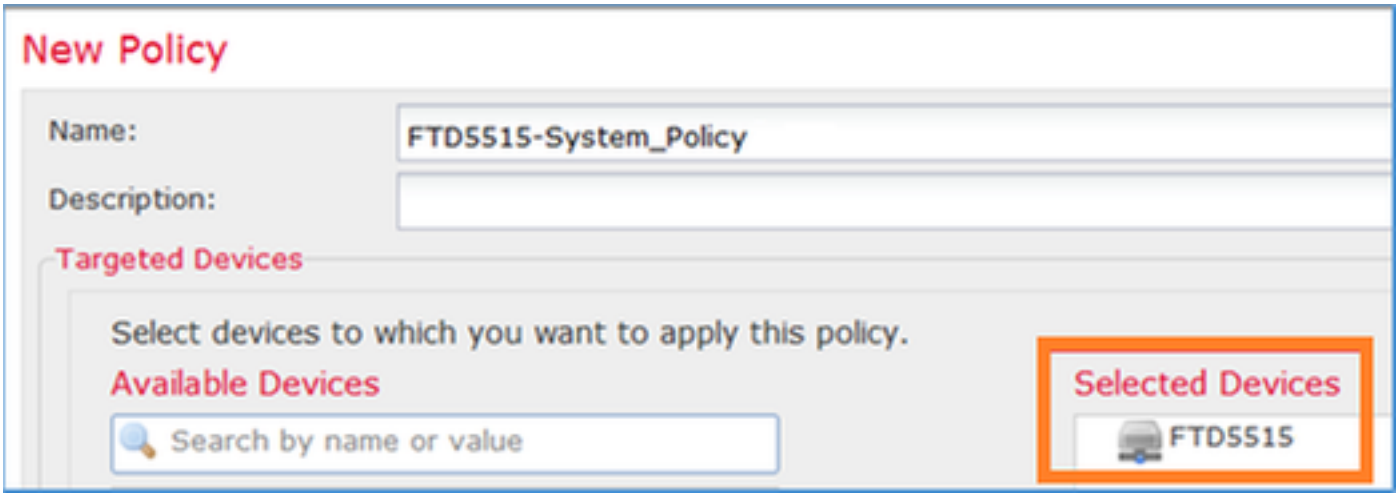
HTTPS 서버는 기본적으로 비활성화되며 액세스가 허용되지 않습니다.

```
> show running-config http
 >
```

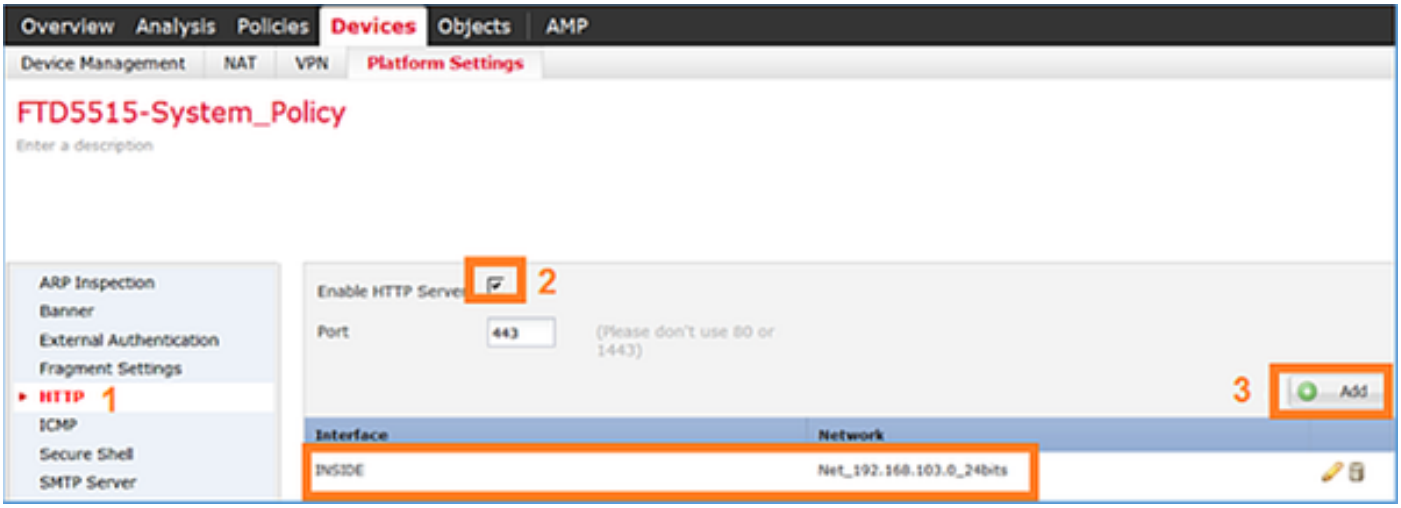
1단계: Devices(디바이스) > Platform Settings(플랫폼 설정)로 이동하여 **New Policy(새 정책)**를 클릭한 다음 **Threat Defense Settings(Threat Defense 설정)**를 선택합니다.



정책 이름과 디바이스 타깃을 지정합니다.



2단계: HTTPS 서버를 활성화하고 HTTPS를 통해 FTD 디바이스 액세스를 허용해야 하는 네트워크를 추가합니다.



Save(저장)하고 **Deploy(구축)**합니다.

정보

정책을 구축하는 동안 **debug http**를 활성화하여 HTTP 서비스가 시작됨을 확인할 수 있습니다.

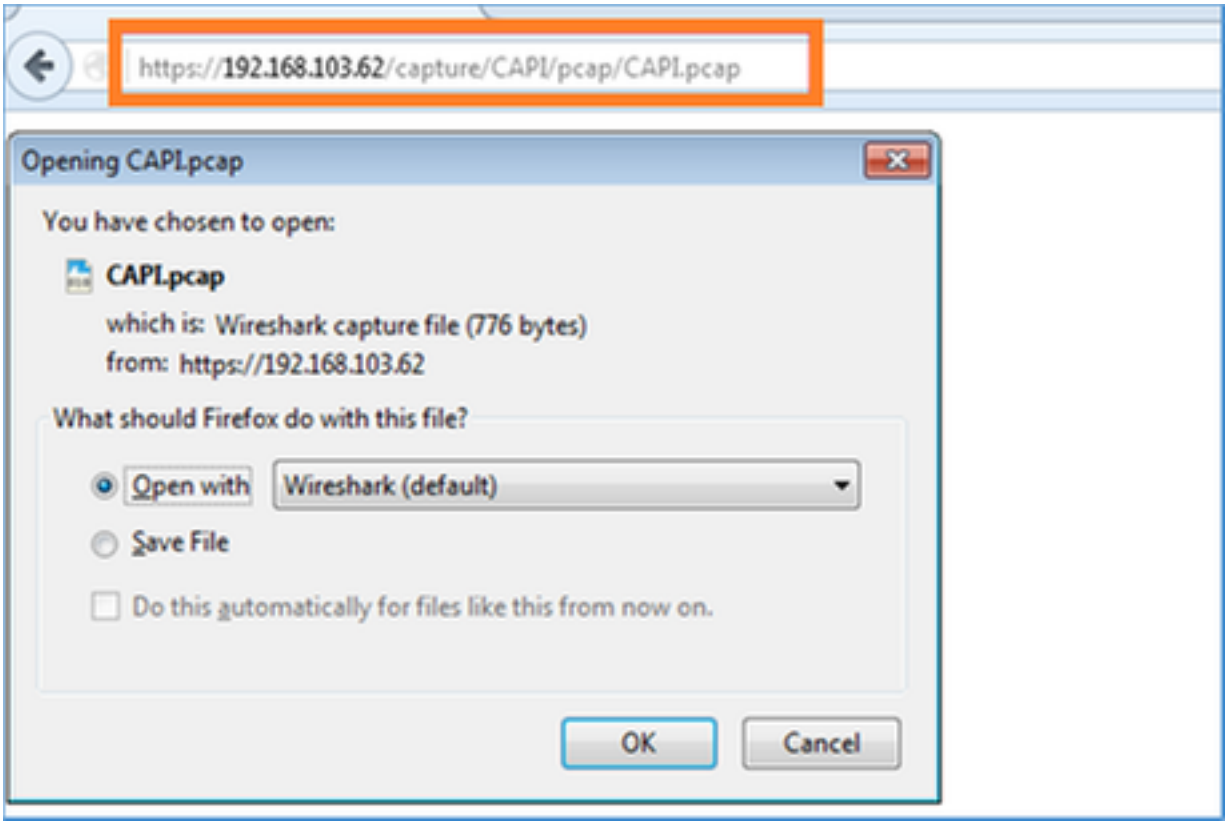
```
> debug http 255 debug http enabled at level 255. http_enable: Enabling HTTP server HTTP server starting.
```

이 경우 FTD CLI에 표시되는 결과는 다음과 같습니다.

```
> unebug all
> show run http http server enable http 192.168.103.0 255.255.255.0 INSIDE
```

호스트-A(192.168.103.1)에서 브라우저를 열고 다음 URL을 사용하여 첫 번째 캡처를 다운로드합니다.

https://192.168.103.62/capture/CAPI/pcap/CAPI.pcap

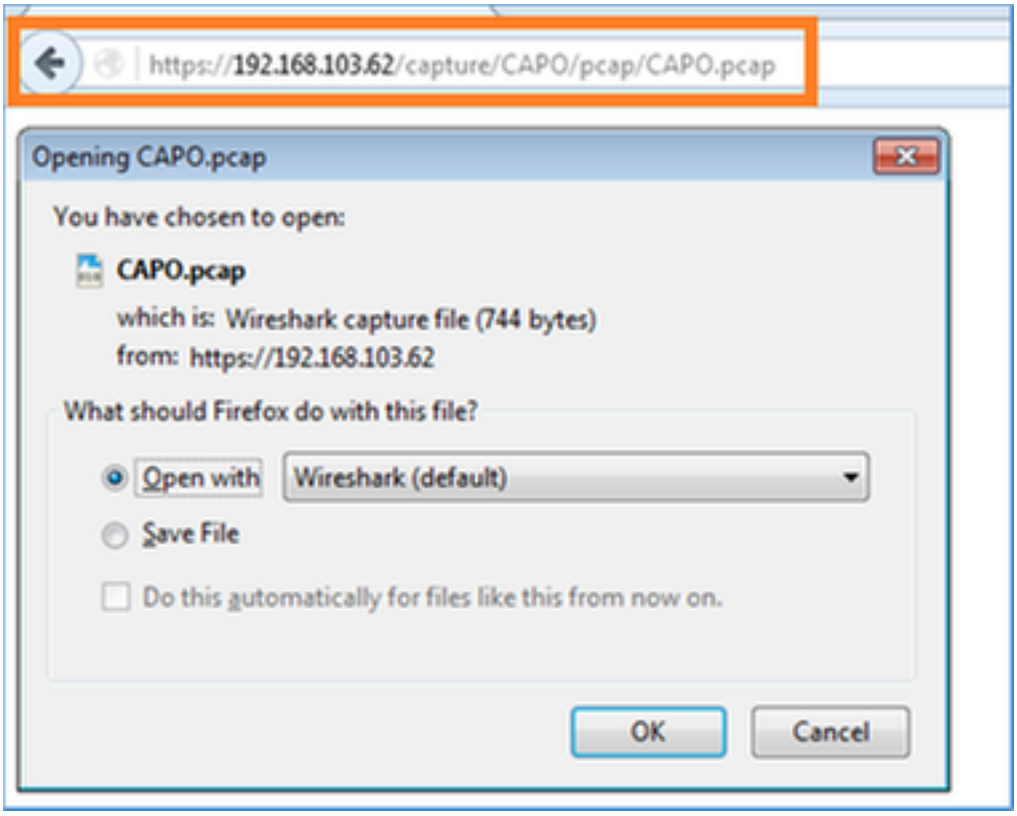


참조용

https://192.168.103.62/capture/CAPI/pcap/CAPI.pcap	HTTP 서버가 활성화된 FTD 데이터 인터페이스의 IP
https://192.168.103.62/capture/CAPI/pcap/CAPI.pcap	FTD 캡처의 이름
https://192.168.103.62/capture/CAPI/pcap/CAPI.pcap	다운로드할 파일의 이름

두 번째 캡처의 경우:

https://192.168.103.62/capture/CAPO/pcap/CAPO.pcap



FTD ASA 엔진 캡처 사용 - FTP/TFTP/SCP를 사용하여 캡처 내보내기

요건

FTP/TFTP/SCP 프로토콜을 사용하여 이전 시나리오에서 가져온 캡처를 내보냅니다.

해결책

FTP 서버로 캡처 내보내기:

```
firepower# copy /pcap capture:CAPI ftp://ftp_username:ftp_password@192.168.78.73/CAPI.pcap
Source capture name [CAPI]? Address or name of remote host [192.168.78.73]? Destination username
[ftp_username]? Destination password [ftp_password]? Destination filename [CAPI.pcap]? !!!!!!
114 packets copied in 0.170 secs
firepower#
```

TFTP 서버로 캡처 내보내기:

```
firepower# copy /pcap capture:CAPI tftp://192.168.78.73 Source capture name [CAPI]? Address or
name of remote host [192.168.78.73]? Destination filename [CAPI]? !!!!!!!!!!!!!!!! 346 packets
copied in 0.90 secs
firepower#
```

SCP 서버로 캡처 내보내기:

```
firepower# copy /pcap capture:CAPI scp://scp_username:scp_password@192.168.78.55 Source capture
name [CAPI]? Address or name of remote host [192.168.78.55]? Destination username
[scp_username]? Destination filename [CAPI]? The authenticity of host '192.168.78.55
(192.168.78.55)' can't be established. RSA key fingerprint is
<cb:ca:9f:e9:3c:ef:e2:4f:20:f5:60:21:81:0a:85:f9:02:0d:0e:98:d0:9b:6c:dc:f9:af:49:9e:39:36:96:33
>(SHA256). Are you sure you want to continue connecting (yes/no)? yes Warning: Permanently added
'192.168.78.55' (SHA256) to the list of known hosts.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 454 packets
copied in 3.950 secs (151 packets/sec)
firepower#
```

FTD ASA 엔진 캡처 사용 - 패킷 추적

요건

다음 필터를 사용하여 FTD에서 캡처를 활성화합니다.

Source IP(소스 IP)	192.168.103.1
Destination IP(목적지 IP)	192.168.101.1
프로토콜	ICMP
인터페이스	내부
Packet tracing(패킷 추적)	예
Number of tracing packets(추적 패킷 수)	100

호스트-A(192.168.103.1) 및 호스트-B(192.168.101.1)에서 ping을 수행하고 캡처를 확인합니다.

해결책

실제 패킷을 추적하면 연결 문제를 트러블슈팅하는 데 매우 유용할 수 있습니다. 즉, 패킷이 통과할 모든 내부 확인 위치를 파악할 수 있습니다. 이렇게 하려면 '**trace detail**' 키워드를 추가하고 추적할 패킷의 양을 지정합니다. 기본적으로 FTD는 처음 50개의 인그레스 패킷을 추적합니다.

이 예에서는 FTD가 내부 인터페이스에서 수신하는 처음 100개 패킷에 대한 추적 세부사항을 사용하여 캡처를 활성화합니다.

```
> capture CAPI2 interface INSIDE trace detail trace-count 100 match icmp host 192.168.103.1 host 192.168.101.1
```

호스트-A에서 호스트-B로의 ping을 수행하고 결과를 확인합니다.

```
C:\Users\cisco>ping 192.168.101.1

Pinging 192.168.101.1 with 32 bytes of data:
Reply from 192.168.101.1: bytes=32 time=2ms TTL=255
Reply from 192.168.101.1: bytes=32 time=2ms TTL=255
Reply from 192.168.101.1: bytes=32 time=2ms TTL=255
Reply from 192.168.101.1: bytes=32 time=8ms TTL=255
```

캡처된 패킷은 다음과 같습니다.

```
> show capture CAPI2 8 packets captured 1: 18:08:04.232989 802.1Q vlan#1577 P0 192.168.103.1 >
192.168.101.1: icmp: echo request 2: 18:08:04.234622 802.1Q vlan#1577 P0 192.168.101.1 >
192.168.103.1: icmp: echo reply 3: 18:08:05.223941 802.1Q vlan#1577 P0 192.168.103.1 >
192.168.101.1: icmp: echo request 4: 18:08:05.224872 802.1Q vlan#1577 P0 192.168.101.1 >
192.168.103.1: icmp: echo reply 5: 18:08:06.222309 802.1Q vlan#1577 P0 192.168.103.1 >
192.168.101.1: icmp: echo request 6: 18:08:06.223148 802.1Q vlan#1577 P0 192.168.101.1 >
192.168.103.1: icmp: echo reply 7: 18:08:07.220752 802.1Q vlan#1577 P0 192.168.103.1 >
192.168.101.1: icmp: echo request 8: 18:08:07.221561 802.1Q vlan#1577 P0 192.168.101.1 >
192.168.103.1: icmp: echo reply 8 packets shown
```

첫 번째 패킷의 추적 결과에서 주목할 만한 부분은 다음과 같습니다.

- '전달 플로우'를 확인할 수 있는 12단계. 이 단계는 ASA 엔진 디스패치 어레이(실제로는 작업의 내부 순서)입니다.
- FTD가 Snort 인스턴스에 패킷을 전송하는 13단계
- Snort 판정이 표시되는 14단계

```
> show capture CAPI2 packet-number 1 trace detail 8 packets captured 1: 18:08:04.232989
000c.2998.3fec a89d.2193.2293 0x8100 Length: 78 802.1Q vlan#1577 P0 192.168.103.1 >
192.168.101.1: icmp: echo request (ttl 128, id 3346) Phase: 1 Type: CAPTURE ... output omitted
... Phase: 12 Type: FLOW-CREATION Subtype: Result: ALLOW Config: Additional Information: New
flow created with id 195, packet dispatched to next module Module information for forward flow
... snp_fp_inspect_ip_options snp_fp_snort snp_fp_inspect_icmp snp_fp_adjacency snp_fp_fragment
snp_ifc_stat Module information for reverse flow ... snp_fp_inspect_ip_options
snp_fp_inspect_icmp snp_fp_snort snp_fp_adjacency snp_fp_fragment snp_ifc_stat Phase: 13 Type:
EXTERNAL-INSPECT Subtype: Result: ALLOW Config: Additional Information: Application: 'SNORT
Inspect' Phase: 14 Type: SNORT Subtype: Result: ALLOW Config: Additional Information: Snort
Verdict: (pass-packet) allow this packet ... output omitted ... Result: input-interface: OUTSIDE
input-status: up input-line-status: up output-interface: OUTSIDE output-status: up output-line-
status: up Action: allow 1 packet shown >
```

FTD 패킷 트레이서 유틸리티 사용

요건

다음 플로우에 대해 패킷 트레이서 유틸리티를 사용하여 패킷이 내부적으로 처리되는 방식을 확인합니다.

Ingress interface(인그레스 인터페이스)	내부
프로토콜	ICMP echo request(ICMP 에코 요청)
Source IP(소스 IP)	192.168.103.1
Destination IP(목적지 IP)	192.168.101.1

해결책

패킷 트레이서는 **가상 패킷**을 생성합니다. 아래에 나와 있는 것처럼 패킷에 대해 Snort 검사가 수행되기는 하지만, Snort 엔진에서 수행되는 캡처에서는 가상 패킷이 실제로는 해당 엔진을 통해 전송되지 않는 것으로 표시됩니다.

```
> packet-tracer input INSIDE icmp 192.168.103.1 8 0 192.168.101.1 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 192.168.101.1 using egress ifc OUTSIDE Phase: 4 Type: ACCESS-LIST
Subtype: log Result: ALLOW Config: access-group CSM_FW_ACL_ global access-list CSM_FW_ACL_
advanced permit ip 192.168.103.0 255.255.255.0 192.168.101.0 255.255.255.0 rule-id 268436482
event-log both access-list CSM_FW_ACL_ remark rule-id 268436482: ACCESS POLICY: FTD5515 -
Mandatory/1 access-list CSM_FW_ACL_ remark rule-id 268436482: L4 RULE: Allow ICMP Additional
Information: This packet will be sent to snort for additional processing where a verdict will be
reached ... output omitted ... Phase: 12 Type: FLOW-CREATION Subtype: Result: ALLOW Config:
Additional Information: New flow created with id 203, packet dispatched to next module Result:
input-interface: INSIDE input-status: up input-line-status: up output-interface: OUTSIDE output-
status: up output-line-status: up Action: allow >
```

관련 문서

Firepower Threat Defense 명령 참조 가이드

Firepower System 릴리스 노트, 버전 6.1.0

Firepower Device Manager, 버전 6.1용 Cisco Firepower Threat Defense 컨피그레이션 가이드