FTD キャプチャおよびパケット トレーサーを操作する場合

目次

概要

この資料に Firepower Threat Defense（FTD）キャプチャおよびパケット トレーサー ユーティリティを使用する方法を記述されています。

パケットキャプチャは最も広く使われたトラブルシューティング ツールの 1 才です。 パケットキャプチャのユース ケースは次のとおりです:

- パケットがデバイスに着くと証明するため
- パケットがデバイスを去ると証明するため
- （例えば ASA ASP をパケットがデバイスによって廃棄されること証明することは廃棄します）

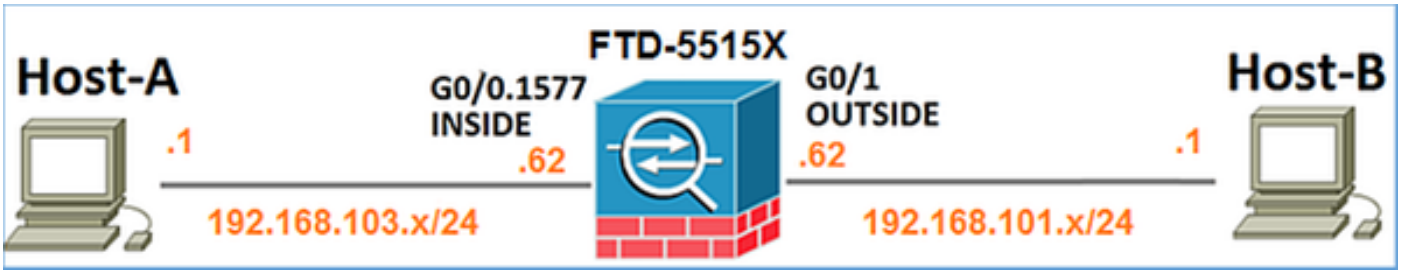
FTD でパケットは 2 つのエンジンによってキャプチャすることができます:

1. ASA エンジン
2. Snort エンジン

使用するコンポーネント

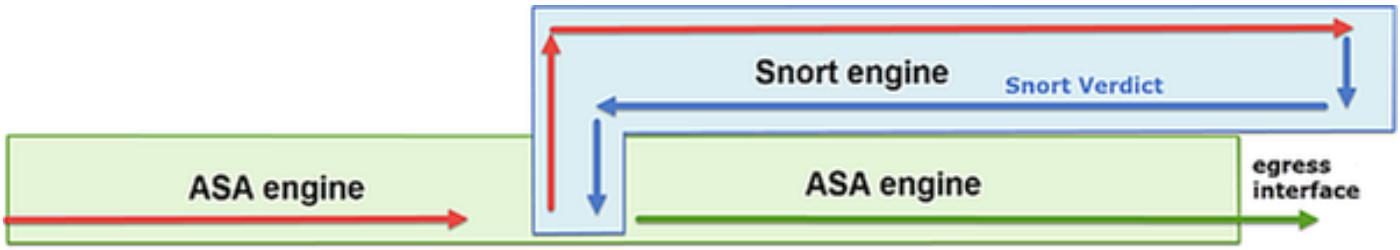
- FTD コード 6.1.0（330）ビルドを実行する ASA5515X
- 6.1.0 を実行する Firepower Management Center（FMC）（330）ビルド

トポロジ



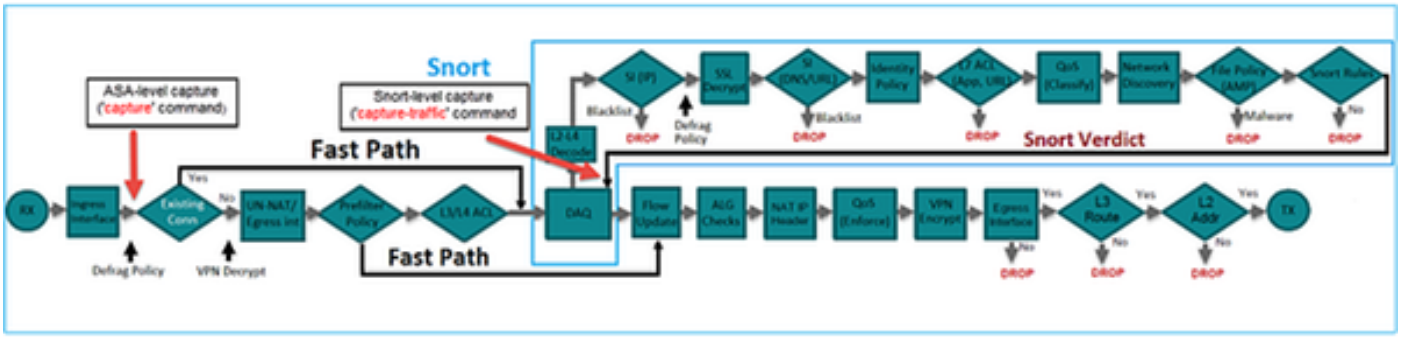
FTD パケット処理

FTD パケット処理は次の通り視覚化することができます:



1. パケットは入力 インターフェイスに入り、ASA エンジンによって処理されます
2. ポリシーが命令すればパケットは Snort エンジンによって検査されます
3. Snort エンジンはパケットのための評決（例えば whitelist、ブラックリスト）を戻します
4. ASA エンジンは Snort の評決に基づいてパケットを廃棄するか、または転送します

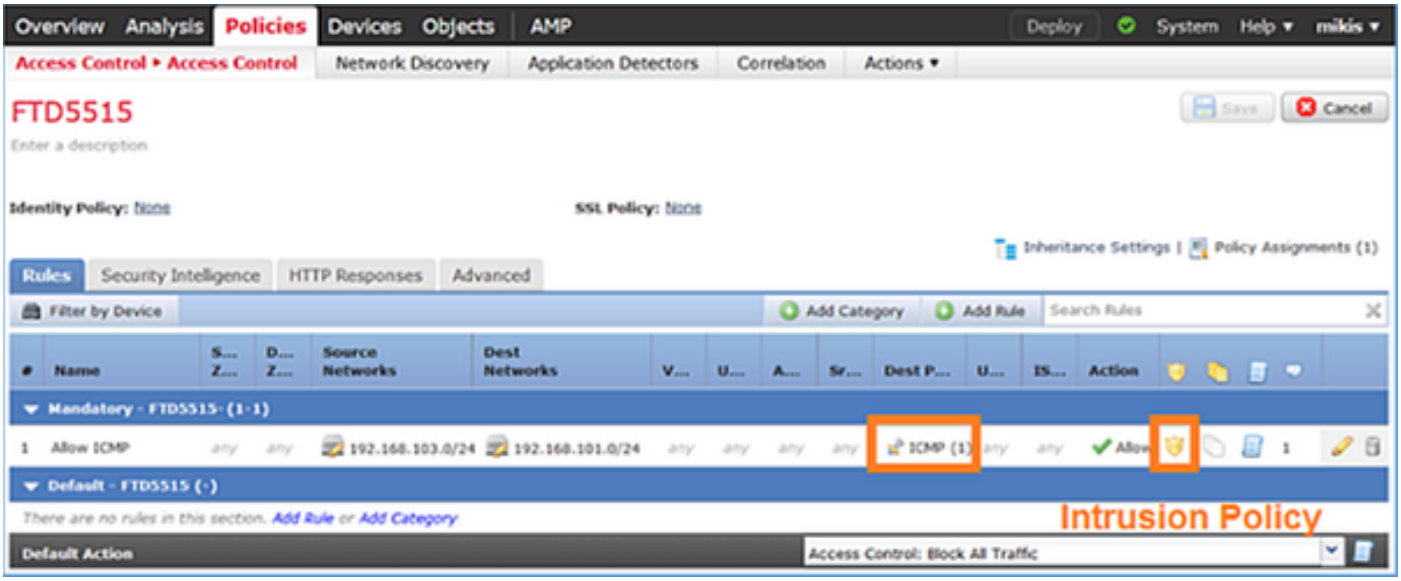
上記のアーキテクチャに基づいて FTD キャプチャは 2 つの異なる場所で奪取 することができます:



Snort エンジン キャプチャとはたらくこと

前提条件

ICMP トラフィックが行くようにする FTD で適用されるアクセス制御ポリシー（ACP）があります。 ポリシーに適用される不正侵入ポリシーがまたあります:



要件

1. フィルタ無しを使用して FTD CLISH モードのイネーブル キャプチャ
2. FTD によって ping し、キャプチャ出力をチェックして下さい

解決策

ステップ 1： FTD コンソールか SSH に br1 インターフェイスにログインし、フィルタを使用して FTD CLISH モードのキャプチャを有効に して下さい

```
> 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.x でコマンドは次のとおりです:

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

ステップ 2： FTD によって ping し、キャプチャ出力をチェックして下さい

```
> 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-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo request, id 0,
seq 1, length 80 12:52:34.749945 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo
reply, id 0, seq 1, length 80 12:52:34.759955 IP olab-vl603-gw.cisco.com > olab-vl647-
gw.cisco.com: ICMP echo request, id 0, seq 2, length 80 12:52:34.759955 IP olab-vl647-
gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 0, seq 2, length 80 12:52:34.759955
IP olab-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo request, id 0, seq 3, length 80
12:52:34.759955 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 0, seq
3, length 80 12:52:34.759955 IP olab-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo
request, id 0, seq 4, length 80 12:52:34.759955 IP olab-vl647-gw.cisco.com > olab-vl603-
gw.cisco.com: ICMP echo reply, id 0, seq 4, length 80
^C <- to exit press CTRL + C
```

Snort エンジン キャプチャとはたらくこと（tcpdump フィルターと）

要件

1. IP 192.168.101.1 のためのフィルタを使用して FTD CLISH モードのキャプチャを有効に して下さい
2. FTD によって ping し、キャプチャ出力をチェックして下さい

解決策

ステップ 1： IP 192.168.101.1 のためのフィルタを使用して FTD CLISH モードのキャプチャを有効に して下さい

```
> 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
```

ステップ 2： FTD によって ping し、出力されるキャプチャをチェックして下さい:

```
> 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
```

数値書式のホストおよびポート番号を見るのに「- n」オプションを使用できます。 たとえば上記のキャプチャはとして示されます:

```
> 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 フィルタ例

例 1

ソース IP か Dst をキャプチャ するため IP = 192.168.101.1 およびソース ポートまたは Dst ポート = TCP/UDP 23:

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

例 2

ソースをキャプチャ するため IP = 192.168.101.1 およびソース ポート = TCP/UDP 23:

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

例 3

ソースをキャプチャ するため IP = 192.168.101.1 およびソース ポート = TCP 23:

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

例 4

ソースを IP = 192.168.101.1 キャプチャ し、パケットの MAC アドレスが「e」オプションを追加するのを見るため:

```
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
```

例 5

10 のパケットをキャプチャした後終了するため:

```
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
```

例 6

キャプチャを名前 capture.pcap のファイルに書き、それをリモートサーバに FTP によってコピーするため:

```
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 エンジン キャプチャとはたらくこと

要件

1. イネーブル次のフィルターを使用して FTD の 2 人のキャプチャ:

送信元 IP	192.168.103.1
宛先 IP	192.168.101.1
プロトコル	ICMP
interface	内部
送信元 IP	192.168.103.1
宛先 IP	192.168.101.1
プロトコル	ICMP
interface	OUTSIDE

2. Host-A （192.168.103.1）から Host-B ping して下さい（192.168.101.1）キャプチャをチェックすれば。

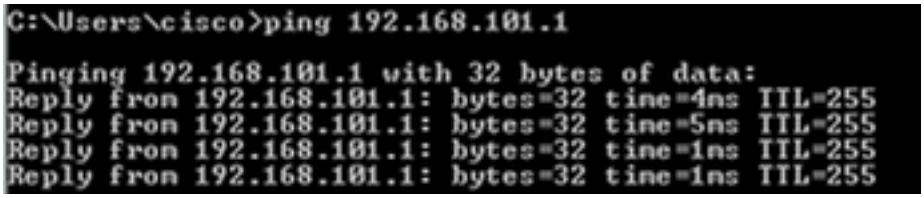
解決策

ステップ 1： キャプチャのイネーブル化:

```
> 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
```

ステップ 2： CLI を使用しているキャプチャのチェック

Host-A からの Host-B への 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
```

2 人のキャプチャに内部インターフェイスの Dot1Q ヘッダによる異なるサイズがあります。 これは次の出力で示すことができます:

```
> 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 エンジン キャプチャとはたらくこと– HTTP を使用しているキャプチャのエクスポート

要件

ブラウザを使用して上記のシナリオで奪取 されるキャプチャを工クスポートして下さい

解決策

そこのブラウザを使用しているキャプチャを工クスポートすることは必要です:

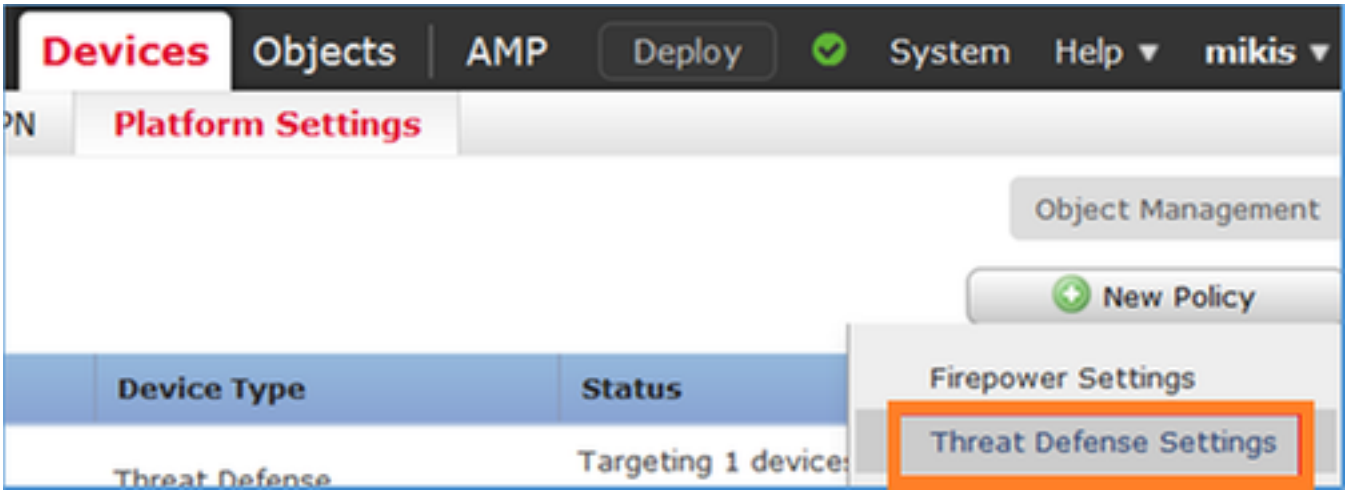
1. イネーブル HTTPS サーバ
2. 割り当て HTTPS アクセス

デフォルトで HTTPS サーバは無効であり、アクセスは許可されません:

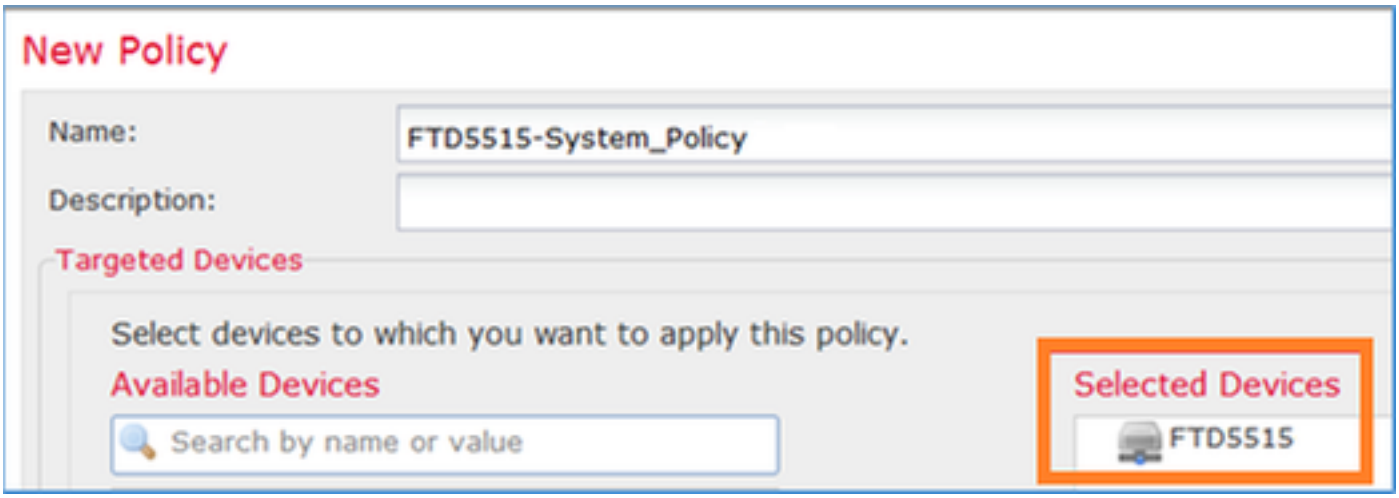
```
> show running-config http

>
```

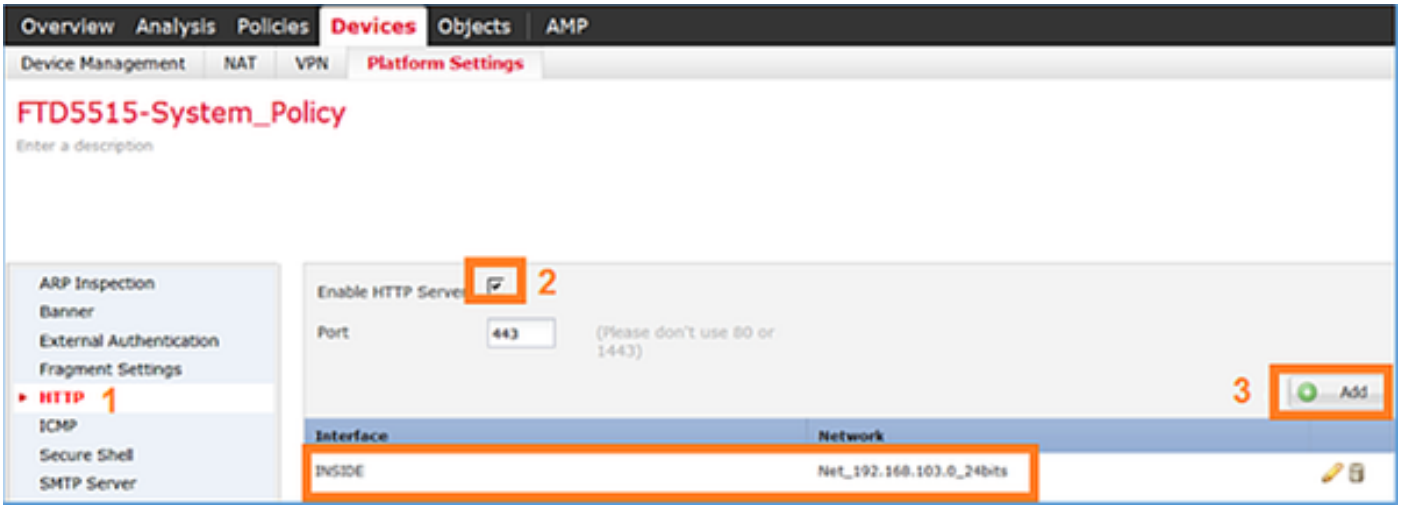
ステップ 1： デバイス > プラットフォーム設定にナビゲートし、**脅威防衛設定を**『New Policy』をクリック し、選択して下さい:



ポリシー名およびデバイスターゲットを規定 して下さい:



ステップ 2 : HTTPS サーバをイネーブルに設定し、HTTPS 上の FTD デバイスにアクセスするようにする必要があるネットワークが追加して下さい:



保存し、展開して下さい

チップ

ポリシーを展開している間 HTTP サービス開始を見ることを**デバッグ http が**可能にすることができます:

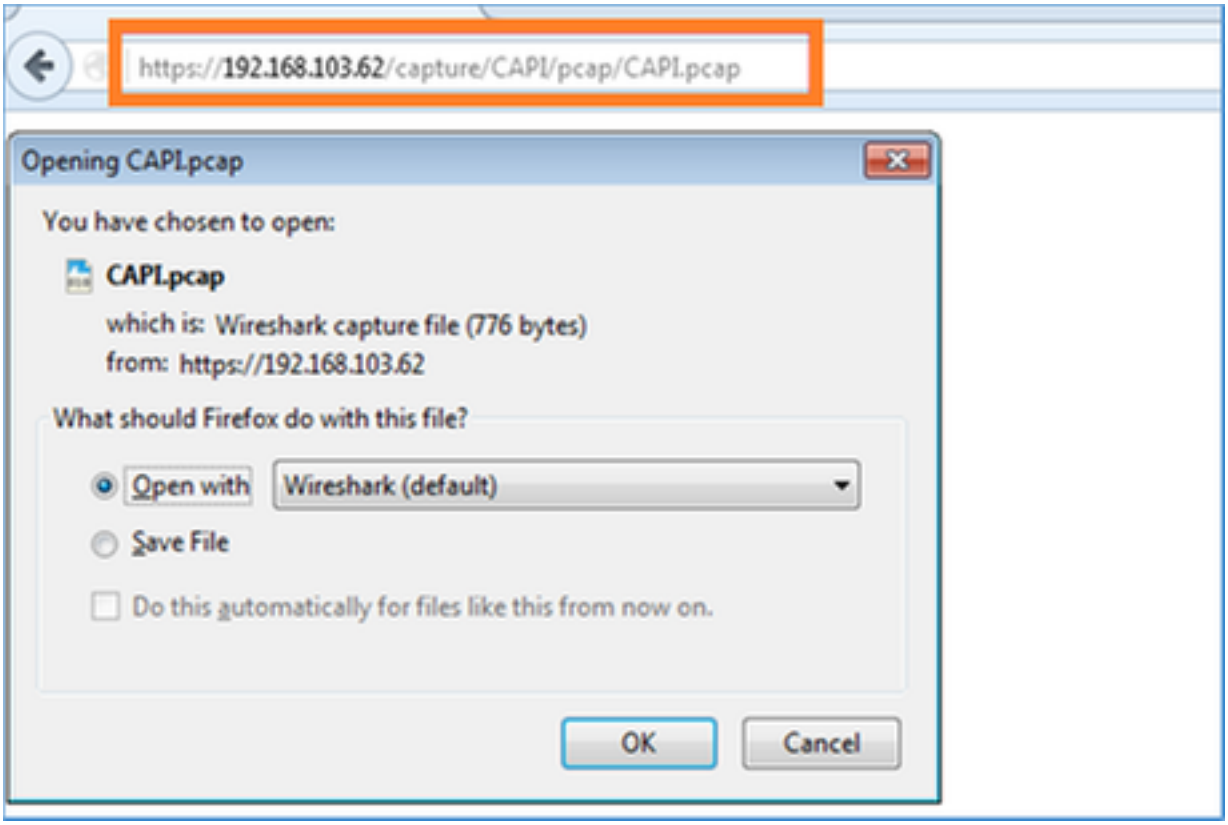
```
> debug http 255
debug http enabled at level 255.
http_enable: Enabling HTTP server HTTP server starting.
```

FTD CLI の結果はここにあります:

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

Host-A のブラウザを開いて下さい (192.168.103.1) 最初のキャプチャをダウンロードするのに次の URL を使用すれば:

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

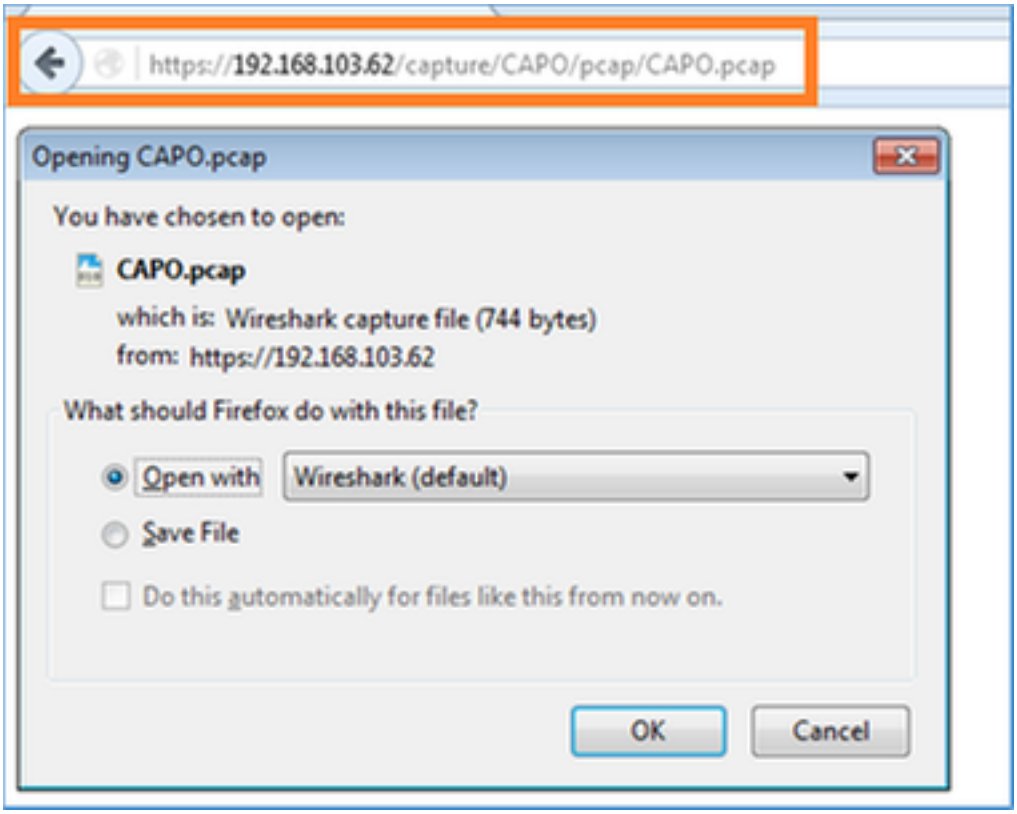


参照に関しては

https://192.168.103.62/capture/CAPI/pcap/CAPI.pcap	HTTPサーバがイネーブルになっている FTD データインターフェイスの IP
https://192.168.103.62/capture/CAPI/pcap/CAPI.pcap	FTD キャプチャの名前
https://192.168.103.62/capture/CAPI/pcap/ CAPI.pcap	ダウンロードされるファイルの名前

第 2 キャプチャに関しては:

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



FTD ASA エンジン キャプチャとはたらくこと– FTP/TFTP/SCP を使用しているキャプチャのエクスポート

要件

FTP/TFTP/SCP プロトコルを使用して上記のシナリオで奪取 されるキャプチャをエクスポートして下さい

解決策

FTP サーバへキャプチャをエクスポートすること:

```
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サーバへキャプチャを工クスポートすること:

```
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 サーバへキャプチャを工クスポートすること:

```
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 エンジン キャプチャとはたらくこと–パケットのトレース

要件

次のフィルターを使用して FTD のキャプチャを有効に して下さい:

送信元 IP	192.168.103.1
宛先 IP	192.168.101.

	1
プロトコル	ICMP
interface	内部
パケットトレース	yes
トレース パケットの数	100

Host-A （ 192.168.103.1 ）から Host-B ping して下さい（ 192.168.101.1 ）キャプチャをチェックすれば。

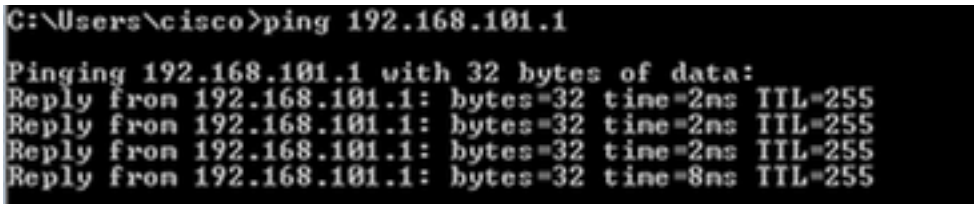
解決策

実パケットをトレースすることは接続上の問題を解決するために非常に役立ちます。 それはパケットが通過しているすべての内部チェックを見ることを割り当てます。 「**トレース詳細**」キーワードを追加し、トレースされるパケットの量を規定 して下さい。 デフォルトで FTD は最初の 50 の入力パケットをトレースします。

この場合 FTD が内部インターフェイスで受信する最初の 100 つのパケットのためのトレース詳細を持つイネーブル キャプチャ:

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

Host-A から Host-B に ping し、結果をチェックして下さい:



キャプチャされるパケットはここにあります:

```
> 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
```

最初のパケットのトレースはここにあります。 興味深い部品は次のとおりです:

- 「前方フロー」が見られる場合があるところフェーズ 12。 これはです ASA エンジン ディスパッチ アレイ（ 効果的に内部オペレーションの順序 ）
- FTD が鼻を鳴らすためにパケットを例 送信 するところフェーズ 13
- Snort 評決が見られるところフェーズ 14

```
> 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 パケット トレーサー ユーティリティの使用

要件

パケット トレーサー ユーティリティを次のフローのために使用し、パケットがどのように内部で処理されるかチェックして下さい:

入力インターフェイス	内部
プロトコル	ICMPエコー要求
送信元 IP	192.168.103.1
宛先 IP	192.168.101.1

解決策

パケット トレーサーは**バーチャル パケットを**生成します。 バーチャル パケットはそれを通して実際に送信 されて いないことがパケットの下でである鼻を鳴らすサブジェクト インスペクション見られる場合があるが Snort エンジンのキャプチャが示すと同時に:

```
> 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

>
```

関連資料

Firepower Threat Defense コマンドレファレンスガイド

Firepower システムリリース注記、バージョン 6.1.0

Firepower デバイスマネージャのための Cisco Firepower Threat Defense コンフィギュレーション ガイド、バージョン 6.1