

# firepower脅威対策のキャプチャとPacket Tracerを使用する

## 内容

---

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[FTD パケット処理](#)

[設定](#)

[ネットワーク図](#)

[Snortエンジンによるキャプチャの処理](#)

[前提条件](#)

[要件](#)

[解決方法](#)

[Snortエンジンによるキャプチャの処理](#)

[要件](#)

[解決方法](#)

[Tcpdump フィルタの例](#)

[FTD LINAエンジンによるキャプチャの処理](#)

[要件](#)

[解決方法](#)

[FTD LINAエンジンによるキャプチャの処理：HTTP経由でのキャプチャのエクスポート](#)

[要件](#)

[解決方法](#)

[FTD LINAエンジンによるキャプチャの処理：FTP/TFTP/SCP経由でのキャプチャのエクスポート](#)

[要件](#)

[解決方法](#)

[FTD LINAエンジンによるキャプチャの処理：実際のトラフィックパケットのトレース](#)

[要件](#)

[解決方法](#)

[6.2以降のFMCソフトウェアバージョンのCapture Tool](#)

[回避策：FTD CLIを使用します](#)

[6.2以降のFMCでの実際のパケットのトレース](#)

[FTD Packet Tracerユーティリティ](#)

[要件](#)

[解決方法](#)

[6.2以降のFMCソフトウェアバージョンのPacket Tracer UIツール](#)

[関連情報](#)

---

はじめに

このドキュメントでは、Firepower脅威対策(FTD)キャプチャおよびPacket Tracerユーティリティの使用方法について説明します。

## 前提条件

### 要件

このドキュメントに関する固有の要件はありません。

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づいています。

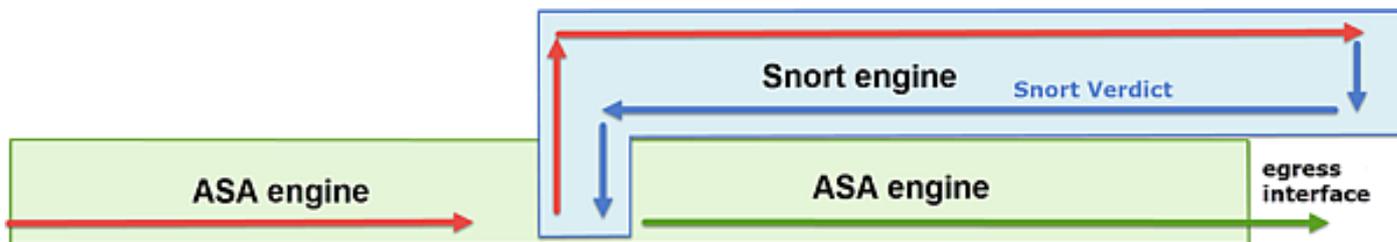
- FTDソフトウェア6.1.0が稼働するASA5515-X
- FTDソフトウェア6.2.2が稼働するFPR4110
- firepower Management Center(FMC)ソフトウェア6.2.2が稼働するFS4000

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな(デフォルト)設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

## 背景説明

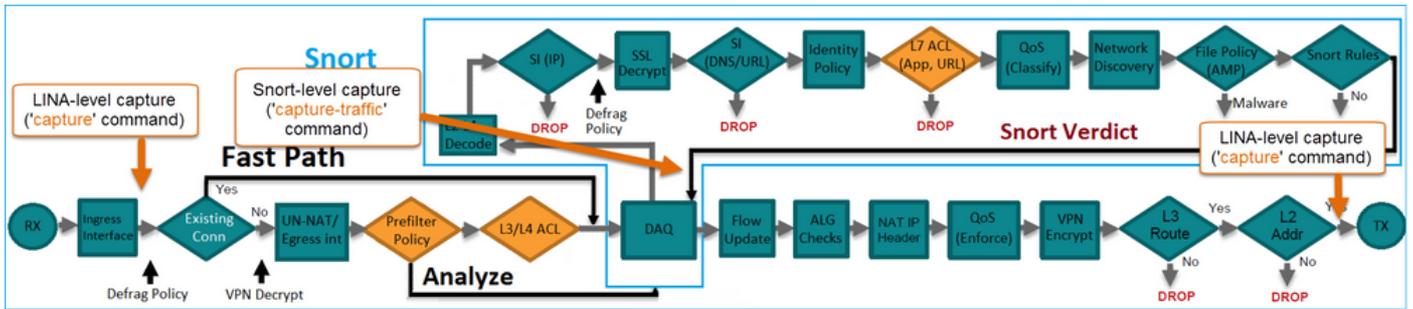
### FTD パケット処理

FTDパケット処理は次のように視覚化されます。



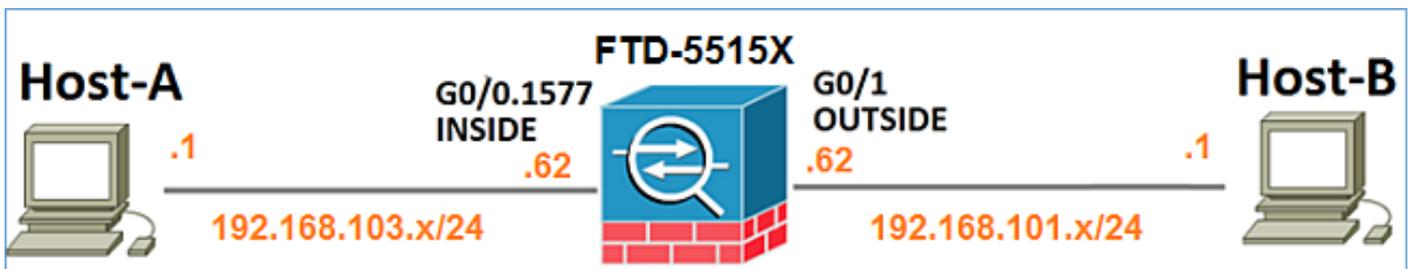
1. パケットが入カインターフェイスに入り、LINAエンジンによって処理されます。
2. Snortエンジンによるパケットの検査がポリシーで必要とされている場合。
3. Snort エンジンがパケットに対する判定を返します。
4. LINA エンジンは、Snort の判定に基づいてパケットをドロップまたは転送する。

アーキテクチャに基づいて、FTDキャプチャは次の場所で取得できます。



## 設定

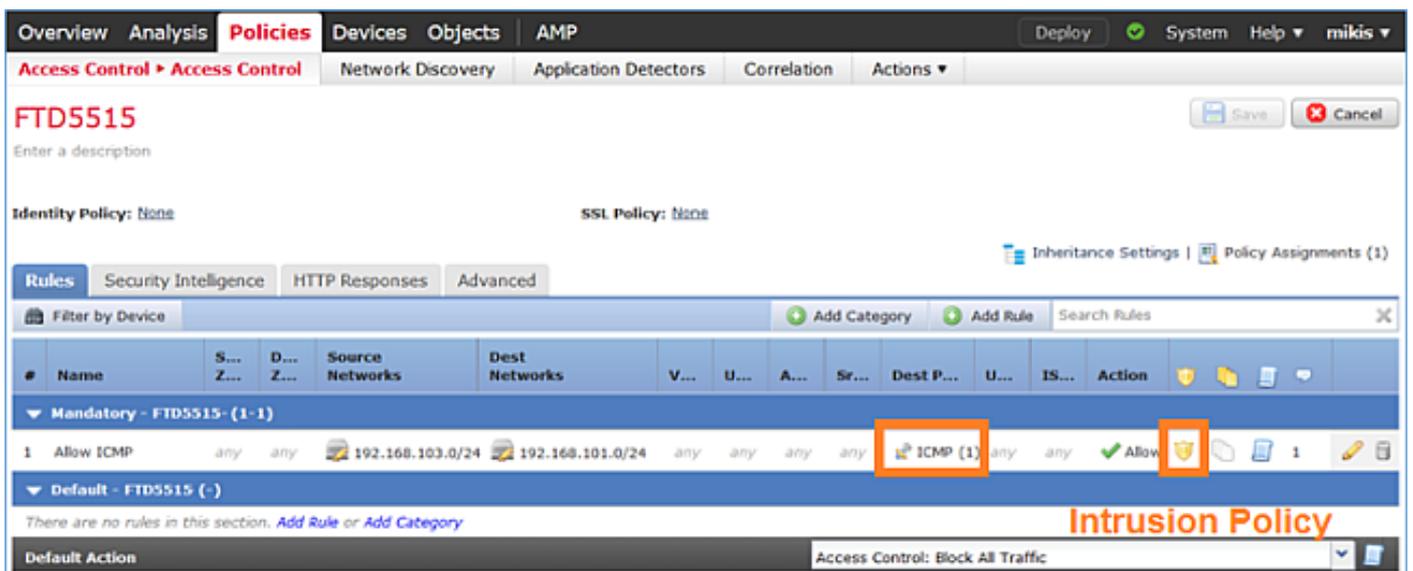
### ネットワーク図



### Snortエンジンによるキャプチャの処理

#### 前提条件

インターネット制御メッセージプロトコル(ICMP)トラフィックの通過を許可するアクセスコントロールポリシー(ACP)がFTDに適用されています。このポリシーには、侵入ポリシーも適用されません。



#### 要件

1. フィルタを使用せずにFTD CLISHモードでキャプチャを有効にします。
2. FTDからpingを実行し、キャプチャされた出力を確認します。

## 解決方法

ステップ 1 : FTDコンソールにログインするか、br1インターフェイスにSSHで接続し、フィルタを使用せずにFTD CLISHモードでキャプチャを有効にします。

```
<#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:
```

FTD 6.0.x では、次のコマンドを使用します。

```
<#root>
```

```
>
```

```
system support
```

```
capture-traffic
```

ステップ 2 : FTDからpingを実行し、キャプチャされた出力を確認します。

```
<#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:

```
12:52:34.749945 IP olab-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo request, id 0, seq 1, length 60
12:52:34.749945 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 0, seq 1, length 60
12:52:34.759955 IP olab-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo request, id 0, seq 2, length 60
12:52:34.759955 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 0, seq 2, length 60
12:52:34.759955 IP olab-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo request, id 0, seq 3, length 60
12:52:34.759955 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 0, seq 3, length 60
12:52:34.759955 IP olab-vl603-gw.cisco.com > olab-vl647-gw.cisco.com: ICMP echo request, id 0, seq 4, length 60
12:52:34.759955 IP olab-vl647-gw.cisco.com > olab-vl603-gw.cisco.com: ICMP echo reply, id 0, seq 4, length 60
^C    <- to exit press CTRL + C
```

## Snortエンジンによるキャプチャの処理

### 要件

1. IP 192.168.101.1のフィルタを使用して、FTD CLISHモードでキャプチャを有効にします。
2. FTDからpingを実行し、キャプチャされた出力を確認します。

### 解決方法

ステップ 1 : IP 192.168.101.1のフィルタを使用して、FTD CLISHモードでキャプチャを有効にします。

```
<#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:
```

```
host 192.168.101.1
```

ステップ 2 : FTDからpingを実行し、キャプチャされた出力を確認します。

```
13:28:36.079982 IP o1ab-v1647-gw.cisco.com > o1ab-v1603-gw.cisco.com: ICMP echo reply, id 3, seq 0, len
13:28:36.079982 IP o1ab-v1647-gw.cisco.com > o1ab-v1603-gw.cisco.com: ICMP echo reply, id 3, seq 1, len
13:28:36.079982 IP o1ab-v1647-gw.cisco.com > o1ab-v1603-gw.cisco.com: ICMP echo reply, id 3, seq 2, len
13:28:36.079982 IP o1ab-v1647-gw.cisco.com > o1ab-v1603-gw.cisco.com: ICMP echo reply, id 3, seq 3, len
13:28:36.079982 IP o1ab-v1647-gw.cisco.com > o1ab-v1603-gw.cisco.com: ICMP echo reply, id 3, seq 4, len
```

-nオプションを使用すると、ホストとポート番号を数値形式で表示できます。たとえば、前述のキャプチャは次のように表示されます。

```
<#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
```

```
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 フィルタの例

例 1 :

送信元IPまたは宛先IP = 192.168.101.1および送信元ポートまたは宛先ポート = TCP/UDP 23をキャプチャするには、次のコマンドを入力します。

```
<#root>
```

```
Options:
```

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

例 2 :

送信元IP = 192.168.101.1および送信元ポート= TCP/UDP 23をキャプチャするには、次のコマンドを入力します。

```
<#root>
```

Options:

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

例 3 :

送信元IP = 192.168.101.1および送信元ポート= TCP 23をキャプチャするには、次のコマンドを入力します。

```
<#root>
```

Options:

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

例 4 :

送信元IP = 192.168.101.1をキャプチャしてパケットのMACアドレスを確認するには、「e」オプションを追加して、次のコマンドを入力します。

```
<#root>
```

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個のパケットをキャプチャした後に終了するには、次のコマンドを入力します。

```
<#root>
```

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 2
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 :

capture.pcapという名前のファイルにキャプチャを書き込んで、FTP経由でリモートサーバにコピーするには、次のコマンドを入力します。

<#root>

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.

>

## FTD LINAエンジンによるキャプチャの処理

### 要件

1.次のフィルタを使用して、FTDで2つのキャプチャを有効にします。

送信元 IP	192.168.103.1
宛先 IP	192.168.101.1
プロトコル	ICMP

インターフェイス	INSIDE
送信元 IP	192.168.103.1
宛先 IP	192.168.101.1
プロトコル	ICMP
インターフェイス	OUTSIDE

2.ホストA(192.168.103.1)からホストB(192.168.101.1)にpingを実行し、キャプチャを確認します。

#### 解決方法

ステップ 1 : キャプチャを有効にします。

<#root>

```
> 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でキャプチャを確認します。

ホスト A からホスト B へ ping を実行します。

```
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=4ms TTL=255
Reply from 192.168.101.1: bytes=32 time=5ms TTL=255
Reply from 192.168.101.1: bytes=32 time=1ms TTL=255
Reply from 192.168.101.1: bytes=32 time=1ms TTL=255
```

<#root>

```
> 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
```

次の出力例に示すように、INSIDEインターフェイスのDot1Qヘッダーにより、2つのキャプチャのサイズが異なります。

<#root>

```
> 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
```

<#root>

```
> 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 LINAエンジンによるキャプチャの処理：HTTP経由でのキャプチャのエクスポート

### 要件

前のシナリオで取得したキャプチャをブラウザでエクスポートします。

## 解決方法

ブラウザでキャプチャをエクスポートするには、次の手順を実行する必要があります。

1. HTTPSサーバを有効にする
2. HTTPS アクセスを許可します。

デフォルトでは、HTTPSサーバは無効で、アクセスは許可されません。

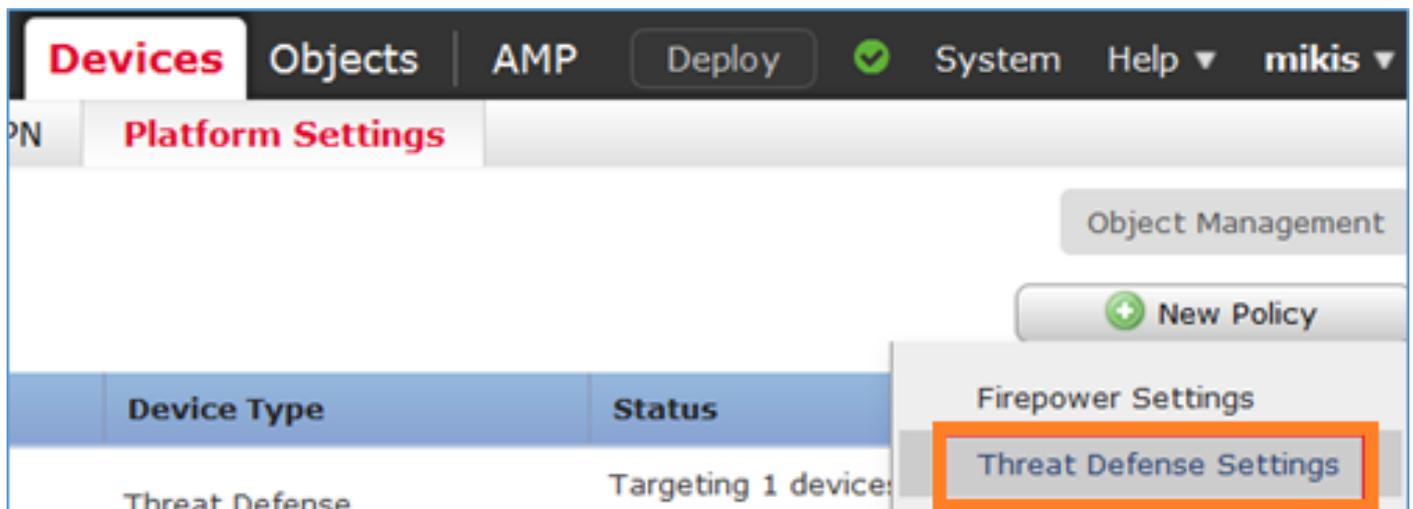
```
<#root>
```

```
>
```

```
show running-config http
```

```
>
```

ステップ 1 : Devices > Platform Settingsの順に移動し、New Policyをクリックして、Threat Defense Settingsを選択します。



ポリシー名と対象デバイスを指定します。

## New Policy

Name:

Description:

**Targeted Devices**

Select devices to which you want to apply this policy.

**Available Devices**

**Selected Devices**

 FTD5515

ステップ 2 : HTTPSサーバを有効にし、FTDデバイスへのHTTPSアクセスを許可するネットワークを追加します。

Overview Analysis Policies **Devices** Objects AMP

Device Management NAT VPN **Platform Settings**

### FTD5515-System\_Policy

Enter a description

- ARP Inspection
- Banner
- External Authentication
- Fragment Settings
- HTTP 1**
- ICMP
- Secure Shell
- SMTP Server

Enable HTTP Server  2

Port  (Please don't use 80 or 1443)

3

Interface	Network
INSIDE	Net_192.168.103.0_24bits

保存して展開します。

ポリシーの導入時に、HTTPサービスの開始を確認するためにdebug httpを有効にできます。

```
<#root>
```

```
> debug http 255
```

```
debug http enabled at level 255.
```

```
http_enable: Enabling HTTP server
HTTP server starting.
```

FTD CLIの結果は次のようになります。

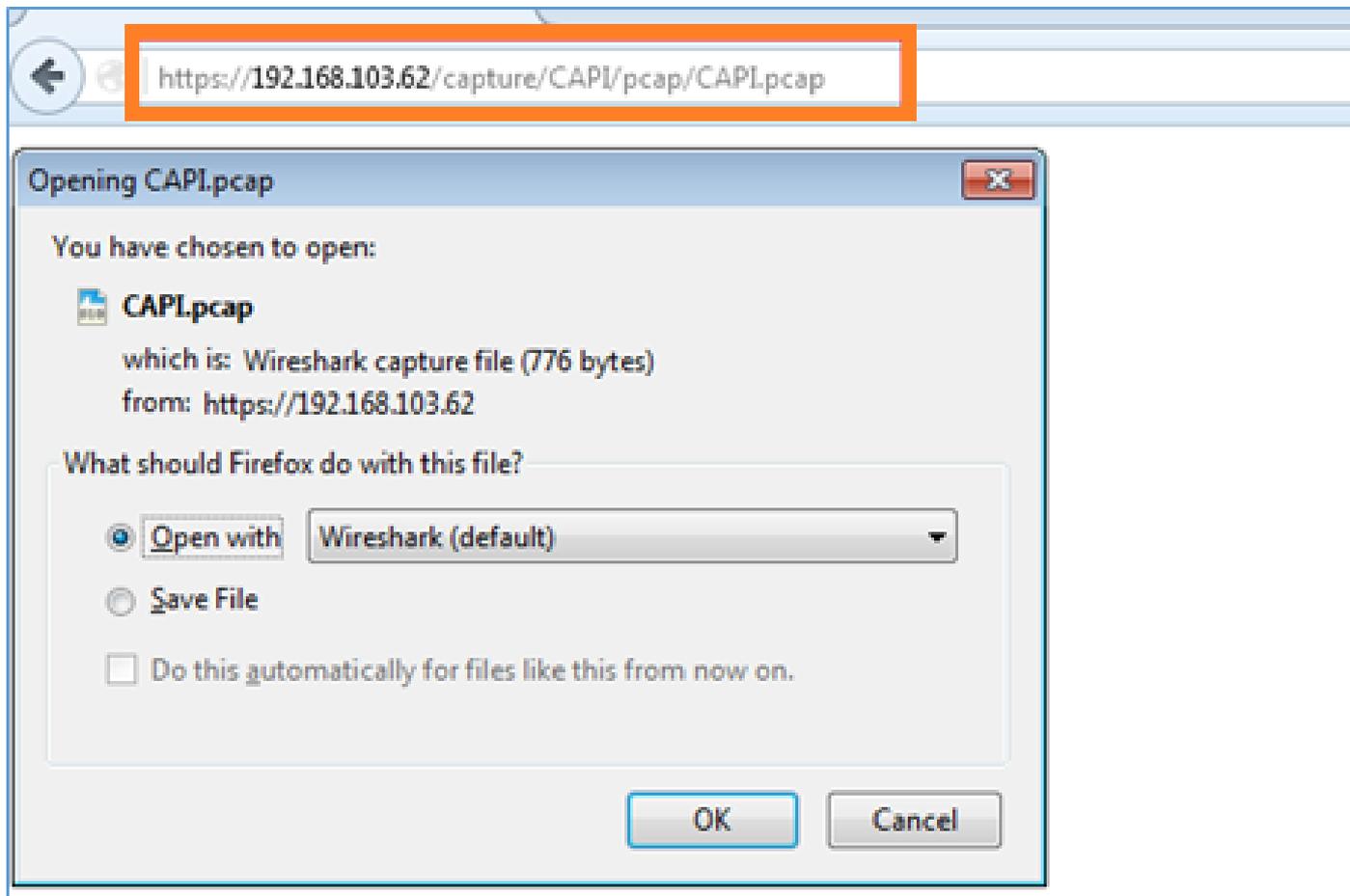
```
<#root>
```

```
> undebug all
```

```
> show run http
```

```
http server enable
http 192.168.103.0 255.255.255.0 INSIDE
```

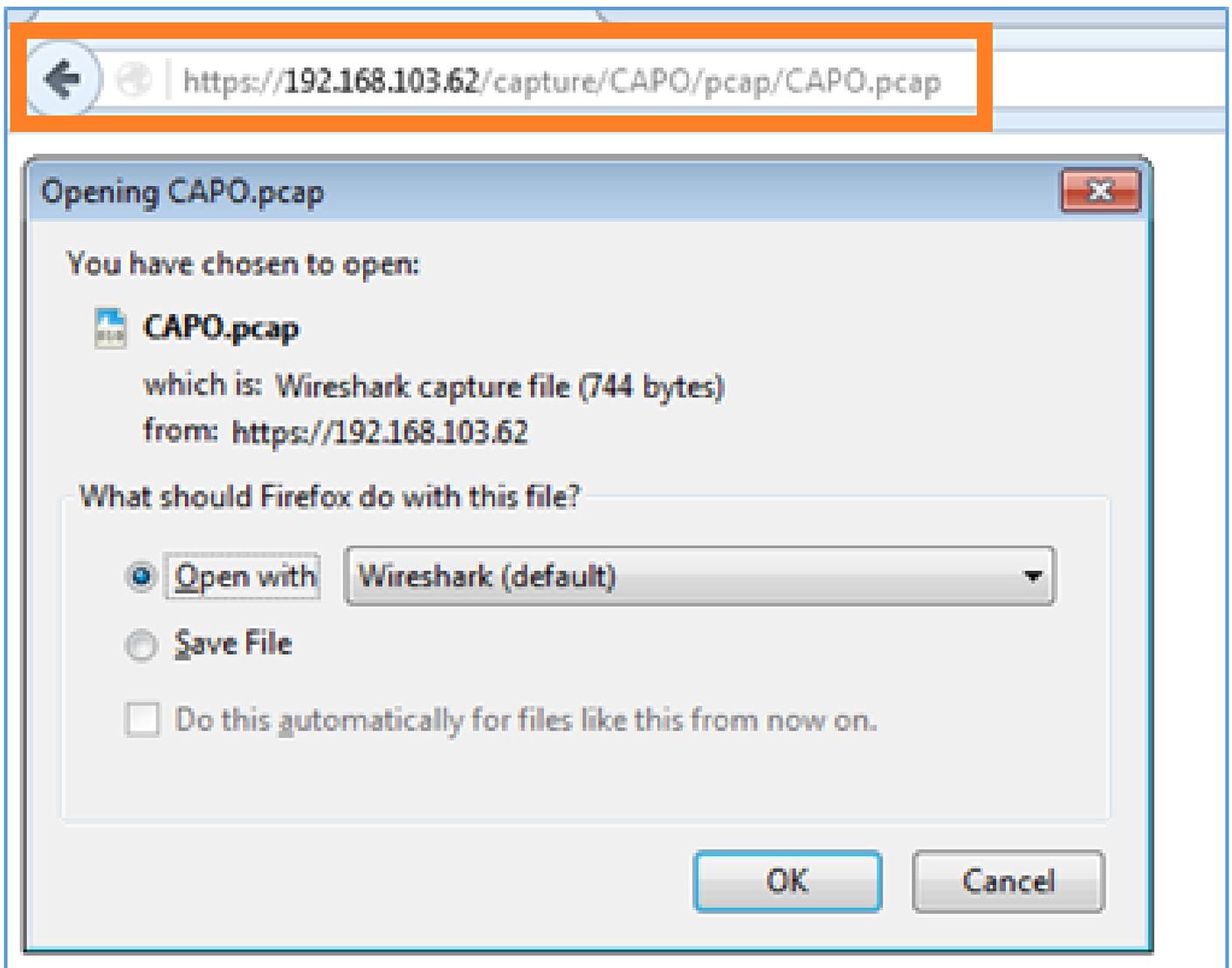
ホストA(192.168.103.1)でブラウザを開き、次のURLを使用して最初のキャプチャ (<https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap>)をダウンロードします。



次のドキュメントを参照してください。

<a href="https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap">https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap</a>	HTTP サーバが有効化されている FTD データ インターフェイスの IP
<a href="https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap">https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap</a>	FTD キャプチャの名前
<a href="https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap">https://192.168.103.62/capture/CAP1/pcap/CAP1.pcap</a>	ダウンロードされるファイルの名 前

2番目のキャプチャには、<https://192.168.103.62/capture/CAPO/pcap/CAPO.pcap>を使用します。



FTD LINAエンジンによるキャプチャの処理：FTP/TFTP/SCP経由での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、前述のシナリオで取得したキャプチャをエクスポートします。

解決方法

キャプチャをFTPサーバにエクスポートします。

```
<#root>
```

```
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]?
```



454 packets copied in 3.950 secs (151 packets/sec)

firepower#

FTDからのオフロードキャプチャ。現在、FTDからキャプチャをオフロードする必要がある場合、最も簡単な方法は次の手順を実行することです。

1. Linaから : `copy /pcap capture:<cap_name> disk0:`
2. FPRルートから - `mv /ngfw/mnt/disk0/<cap_name> /ngfw/var/common/`
3. FMCのUIでSystem > Health > Monitor > Device > Advanced Troubleshootingの順に選択し、フィールドに<cap\_name>を入力してダウンロードします。

FTD LINAエンジンによるキャプチャの処理：実際のトラフィックパケットのトレース

要件

次のフィルタを使用して、FTDでキャプチャを有効にします。

送信元 IP	192.168.103.1
宛先 IP	192.168.101.1
プロトコル	ICMP
インターフェイス	INSIDE
パケットのトレース	yes
トレースするパケット数	100

ホスト A ( 192.168.103.1 ) からホスト B ( 192.168.101.1 ) に ping を実行し、キャプチャ内容を確認します。

解決方法

実際のパケットをトレースすることは、接続問題のトラブルシューティングに非常に役立ちます。これにより、パケットが通過するすべての内部チェックを確認できます。trace detailキーワードを追加して、トレースするパケットの数を指定します。デフォルトでは、FTDは最初の50個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パケットに対して、トレース詳細を使用してキャプチャを有効にします。

```
<#root>
```

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

ホスト A からホスト B に ping を実行し、結果を確認します。

```
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
```

キャプチャされたパケットは次のとおりです。

```
<#root>
```

```
> 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
```

次の出力は、最初のパケットのトレースを示しています。対象となる部分は次のとおりです。

- フェーズ12では「フォワードフロー」が確認されます。これはLINAエンジンのディスパッチアレイです（実質的には内部操作の順序）。
- フェーズ13では、FTDがSnortインスタンスにパケットを送信します。
- フェーズ14では、Snort判定が行われます。

```
<#root>
```

```
> 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 PO 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 ...

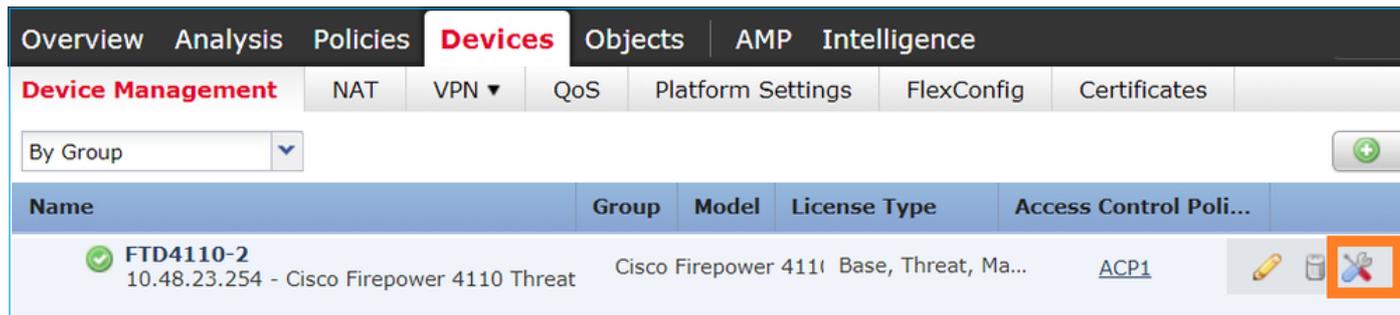
Result:
input-interface: OUTSIDE
input-status: up
input-line-status: up
output-interface: OUTSIDE
output-status: up
output-line-status: up
Action: allow
```

1 packet shown

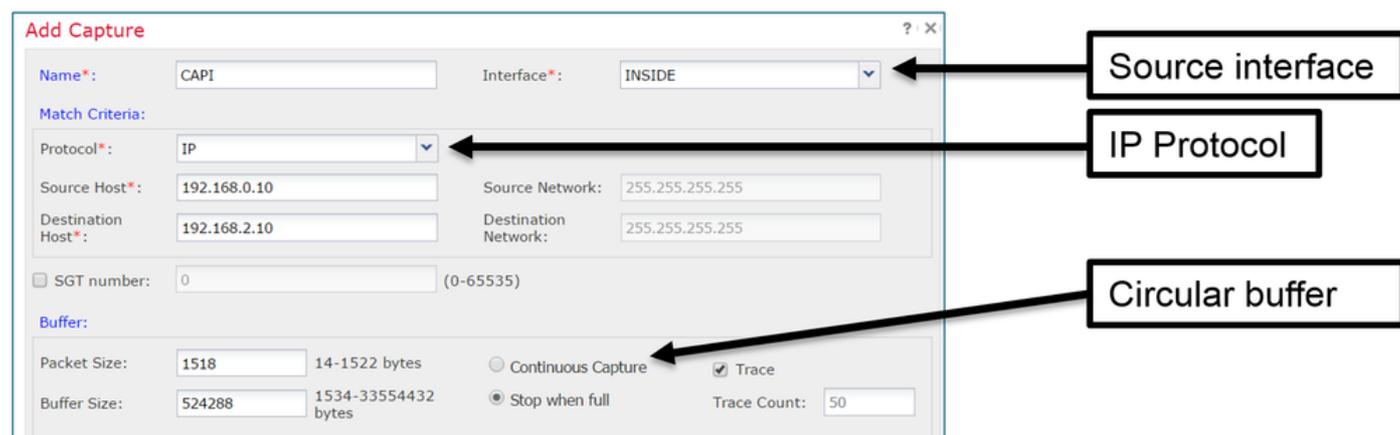
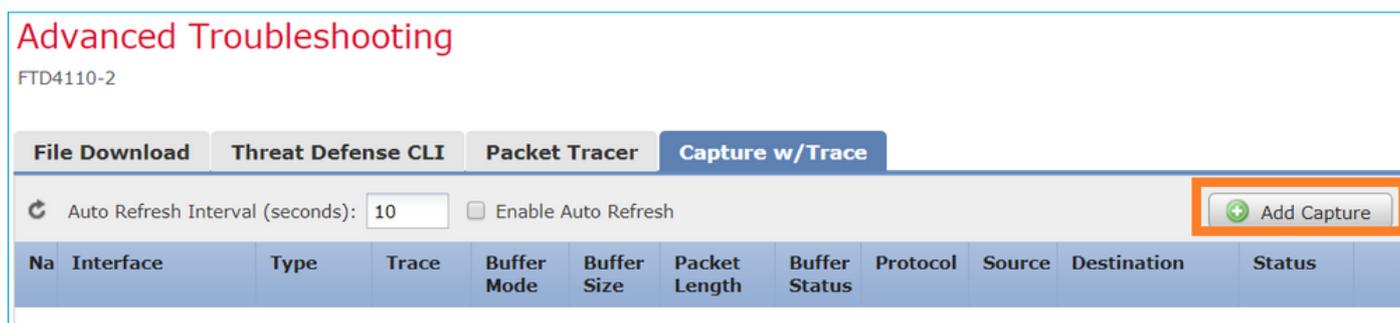
>

## 6.2以降のFMCソフトウェアバージョンのCapture Tool

FMCバージョン6.2.xでは、新しいパケットキャプチャウィザードが導入されました。Devices > Device Managementの順に移動し、Troubleshootアイコンをクリックします。次に、Advanced Troubleshootingを選択し、最後にCapture w/Traceを選択します。



Add Captureを選択して、FTDキャプチャを作成します。

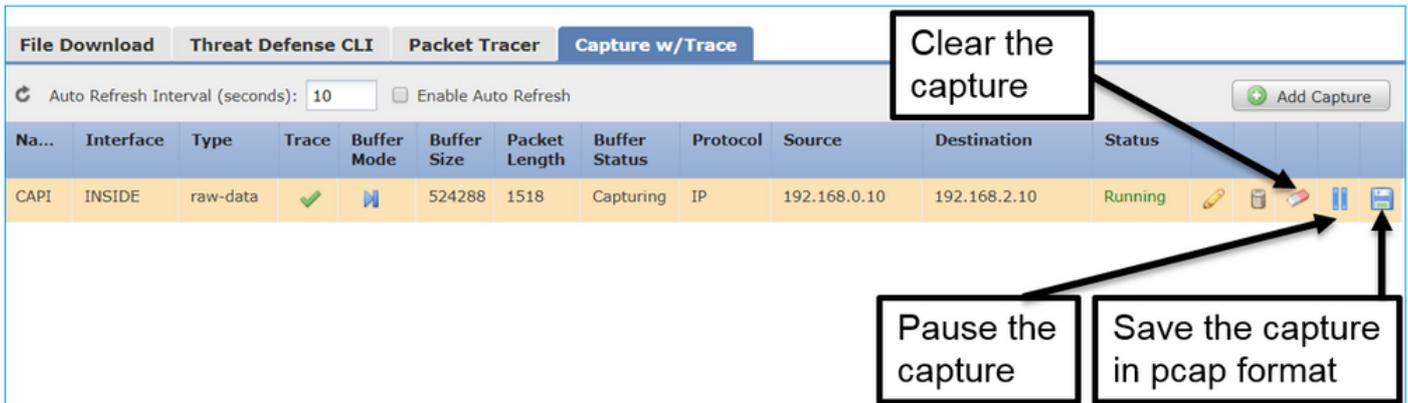


現在のFMC UIの制限は次のとおりです。

- 送信元ポートと宛先ポートを指定できません
- 基本的なIPプロトコルのみ照合できます
- LINAエンジンのASPドロップのキャプチャを有効にできません

回避策：FTD CLIを使用します

FMC UIからキャプチャを適用するとすぐに、キャプチャが実行されます。



FTD CLIのキャプチャ：

<#root>

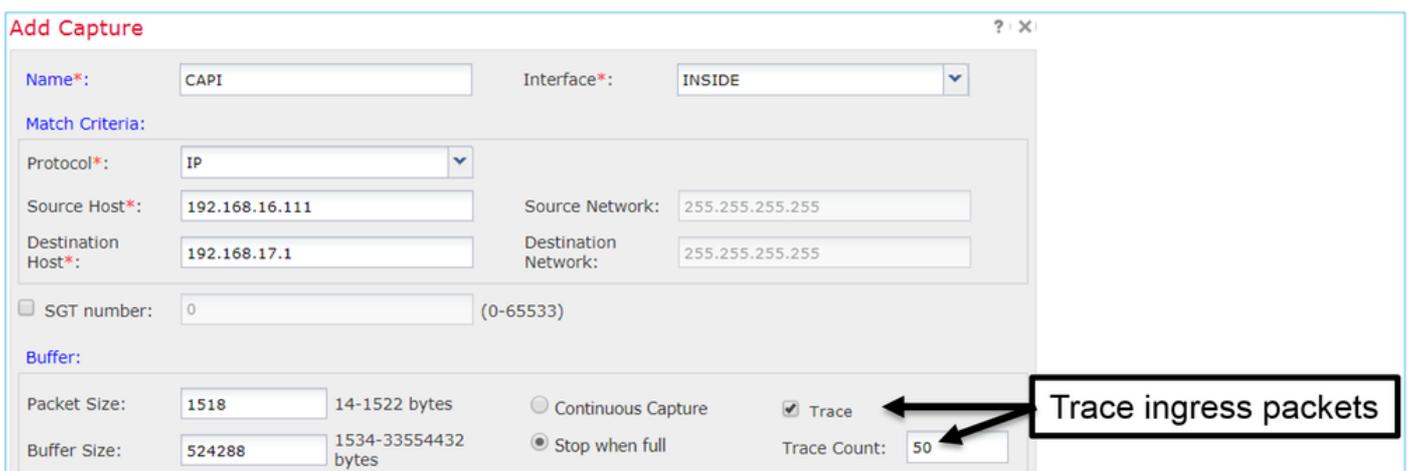
> show capture

```
capture CAPI%intf=INSIDE% type raw-data trace interface INSIDE [Capturing - 0 bytes]
  match ip host 192.168.0.10 host 192.168.2.10
```

>

## 6.2以降のFMCでの実際のパケットのトレース

FMC 6.2.xでは、Capture w/Traceウィザードを使用してFTD上の実際のパケットをキャプチャおよびトレースできます。



トレースされたパケットはFMCのUIで確認できます。

## Advanced Troubleshooting

FTD4110-2

The screenshot shows the Packet Tracer interface with the 'Capture w/Trace' tab selected. The capture table shows a single capture on the 'INSIDE' interface, type 'raw-data', with a status of 'Running'. Below the table, the packet details are displayed, including the Snort verdict: 'Verdict PASS'.

```
Config-
Additional Information:
New flow created with id 78, packet dispatched to next module

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 Trace:
Packet: ICMP
AppID: service ICMP (3501), application unknown (0)
Firewall: allow rule, 'Default Action', allow
NAP id 1, IPS id 2, Verdict PASS
Snort Verdict: (pass-packet) allow this packet
```

The packet is traced

The Snort verdict

## FTD Packet Tracerユーティリティ

### 要件

このフローにPacket Tracerユーティリティを使用して、パケットが内部でどのように処理されるかを確認します。

入カインターフェイス	INSIDE
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

### 解決方法

Packet Tracerは仮想パケットを生成します。この例に示すように、パケットはSnortインスペクションの対象になります。Snortレベルで同時に取得されたキャプチャ(capture-traffic)は、ICMPエコー要求を示します。

<#root>

> 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  
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 is sent to snort for additional processing where a verdict is 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

Phase: 13  
Type: SNORT  
Subtype:  
Result: ALLOW  
Config:  
Additional Information:  
Snort Trace:  
Packet: ICMP

```
AppID: service ICMP (3501), application unknown (0)
Firewall: allow rule, id 268440225, allow
NAP id 2, IPS id 0, Verdict PASS
Snort Verdict: (pass-packet) allow this packet
```

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

```
>
```

パケットトレーサテスト時のSnortレベルのキャプチャは、仮想パケットを示します。

```
<#root>
```

```
>
```

```
capture-traffic
```

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

- 0 - management0
- 1 - Router

```
Selection? 1
```

```
Please specify tcpdump options desired.
(or enter '?' for a list of supported options)
```

```
Options:
```

```
-n
```

```
13:27:11.939755 IP 192.168.103.1 > 192.168.101.1: ICMP echo request, id 0, seq 0, length 8
```

## 6.2以降のFMCソフトウェアバージョンのPacket Tracer UIツール

FMCバージョン6.2.xでは、Packet Tracer UIツールが導入されました。このツールはキャプチャツールと同じ方法でアクセスでき、FMC UIからFTD上でPacket Tracerを実行できます。

Configuration Users Domains Integration Updates Licenses Health Monitor

## Advanced Troubleshooting

FTD4110-2

File Download Threat Defense CLI **Packet Tracer** Capture w/Trace

Select the packet type and supply the packet parameters. Click start to trace the packet.

Packet type:	TCP	Interface*:	INSIDE
Source*:	IP address (IPv4) 192.168.0.10	Source Port*:	1111
Destination*:	IP address (IPv4) 192.168.2.10	Destination Port*:	http
SGT number:	SGT number. (0-65533)	VLAN ID:	VLAN ID... (1-4096)
Output Format:	summary	Destination Mac Address:	XXXX.XXXX.XXXX

Start Clear

Output

```
Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
MAC Access list
Phase: 2
```

The source interface

The tracer output

## 関連情報

- [Firepower Threat Defense コマンド リファレンス ガイド](#)
- [Firepower システム リリース ノート、バージョン 6.1.0](#)
- [Cisco Firepower Threat Defense バージョン 6.1 コンフィギュレーション ガイド \( Firepower Device Manager 用 \)](#)
- [テクニカル サポートとドキュメント - Cisco Systems](#)

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。