

ネットワークの問題をトラブルシューティングするためのFirepowerファイアウォールキャプチャの分析

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](#)

[NGFW 製品ファミリでキャプチャを収集およびエクスポートする方法](#)

[EXOS キャプチャの収集](#)

[FTD.LINA キャプチャの有効化および収集](#)

[FTD Snort キャプチャの有効化および収集](#)

[トラブルシューティング](#)

[Case 1. 出カインターフェイスにTCP SYNなし](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[考えられる原因と推奨されるアクションの概要](#)

[Case 2. クライアントからのTCP SYN、サーバからのTCP RST](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[Case 3.1 つのエンドポイントからのTCP 3ウェイハンドシェイク+ RST](#)

[キャプチャ分析](#)

[3.1: クライアントからの TCP 3 ウェイハンドシェイク + 遅延 RST](#)

[推奨される対処法](#)

[3.2: クライアントからの TCP 3 ウェイハンドシェイク + 遅延 FIN/ACK + サーバーからの遅延 RST](#)

[推奨される対処法](#)

[3.3: クライアントからの TCP 3 ウェイハンドシェイク + 遅延 RST](#)

[推奨される対処法](#)

[3.4: サーバーからの TCP 3 ウェイハンドシェイク + 即時 RST](#)

[推奨される対処法](#)

[ケース 4. クライアントからのTCP RST](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[ケース 5. 遅いTCP転送 \(シナリオ1\)](#)

[シナリオ 1. 遅い転送](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[シナリオ 2. 高速転送](#)

[Case 6. 遅いTCP転送 \(シナリオ2\)](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[キャプチャをエクスポートして、入力パケットと出力パケットの時間差をチェックします。ケース7.TCP接続の問題 \(パケット破損\)](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[Case 8.UDP接続の問題 \(欠落パケット\)](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[Case 10.HTTPS接続の問題 \(シナリオ2\)](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[Case 12.断続的な接続の問題 \(ARPポイズニング\)](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[Case 13.CPU Hogを引き起こすSNMPオブジェクトID\(OID\)の特定](#)

[キャプチャ分析](#)

[推奨される対処法](#)

[関連情報](#)

はじめに

このドキュメントでは、ネットワークの問題を効果的にトラブルシューティングするための、さまざまなパケットキャプチャ分析手法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Firepower プラットフォーム アーキテクチャ
- NGFW ログ
- NGFW パケットトレーサ

さらに、パケットキャプチャの分析を開始する前に、次の要件を満たすことを強くお勧めします。

- プロトコルの動作を把握する：キャプチャされたプロトコルの動作を理解できない場合は、パケットキャプチャのチェックを開始しないでください。
- トポロジを把握する：中継デバイスをエンドツーエンドで把握する必要があります。これが不可能な場合は、少なくともアップストリームデバイスとダウンストリームデバイスを知っている必要があります。
- アプライアンスの把握：デバイスでのパケットの処理方法、関連するインターフェイス (入力/出力)、デバイスアーキテクチャ、さまざまなキャプチャポイントを把握しておく必要があります。

- 設定の理解：デバイスによってパケットフローが処理される方法を、次の観点から理解している必要があります。
 - ルーティング/出カインターフェイス
 - 適用されているポリシー
 - ネットワーク アドレス変換 (NAT)
- 使用可能なツールを把握する：キャプチャに加えて、他のツールや手法 (ログイングやトレーサなど) を適用する準備を整え、必要に応じてキャプチャされたパケットと関連付けることを推奨します。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- FTD ソフトウェア 6.5.x を実行している FP4140 (ほとんどのシナリオがこれに基づいています)
- FTD ソフトウェア 6.5.x を実行している FMC

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

パケットキャプチャは、現在利用可能なトラブルシューティング ツールのなかで最も見過ごされているものの一つです。Cisco TACは、キャプチャされたデータの分析を通じて、多くの問題を解決します。

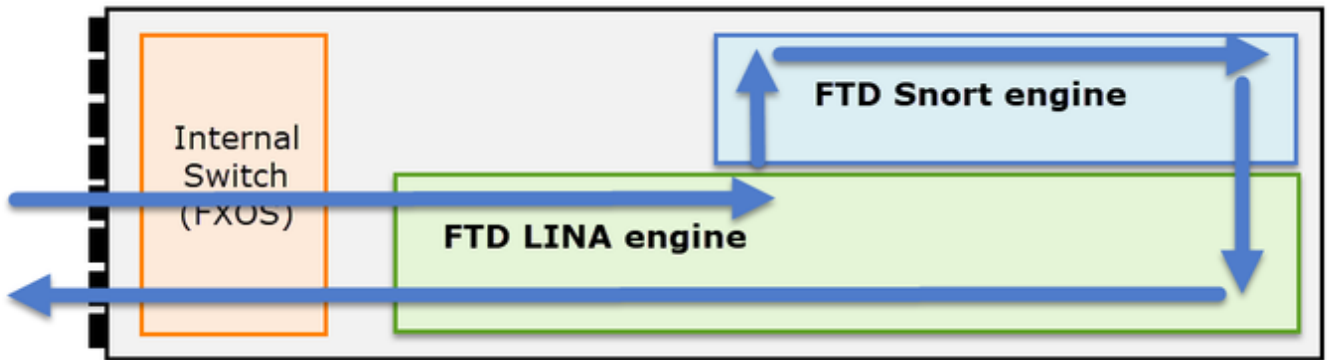
このドキュメントの目的は、ネットワークエンジニアとセキュリティエンジニアが、主にパケットキャプチャ分析に基づいて一般的なネットワークの問題を特定し、トラブルシューティングできるようにすることです。

このドキュメントに示されているすべてのシナリオは、Cisco Technical Assistance Center (TAC) で確認された実際のユーザーの事例に基づいています。

このドキュメントでは、シスコ次世代ファイアウォール (NGFW) の観点でのパケットキャプチャについて説明していますが、同じ概念が他のデバイスタイプにも適用されます。

NGFW 製品ファミリでキャプチャを収集およびエクスポートする方法

Firepower アプライアンス (1xxx、21xx、41xx、93xx) と Firepower Threat Defense (FTD) アプリケーションを使用している場合、パケット処理を図示すると次のようになります。



1. パケットが入カインターフェイスに入り、シャーシの内部スイッチによって処理されます。
2. パケットが、主に L3/L4 チェックを行う FTD LINA エンジンに入ります。
3. ポリシーで要求される場合、パケットが Snort エンジンによって検査されます (主に L7 検査)。
4. Snort エンジンがパケットに対する判定を返します。
5. 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。
6. パケットがシャーシの内部スイッチを通過してシャーシから出ます。

示されているアーキテクチャに基づいて、FTDキャプチャは3つの異なる場所で取得できます。

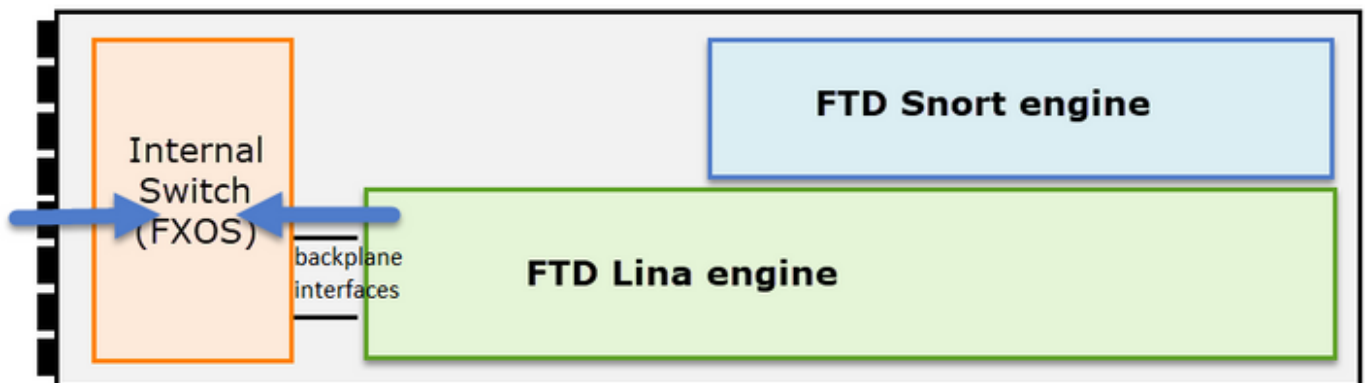
- FXOS
- FTD LINA エンジン
- FTD Snort エンジン

FXOS キャプチャの収集

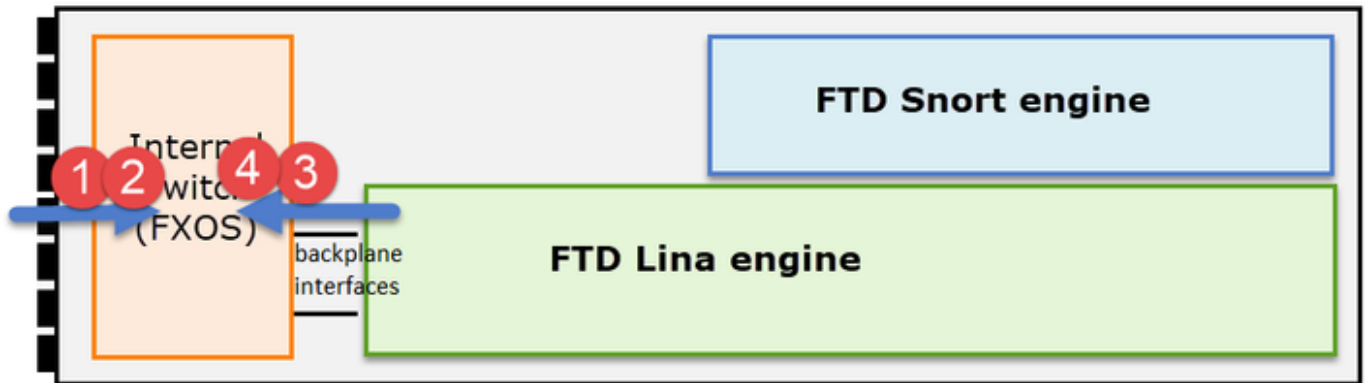
このプロセスについては、次のドキュメントで説明されています。

https://www.cisco.com/c/en/us/td/docs/security/firepower/fxos/fxos271/web-guide/b_GUI_FXOS_ConfigGuide_271/troubleshooting.html#concept_E8823CC63C934A909BBC0DF12F


次の図のように、FXOS キャプチャは、内部スイッチの観点から入力方向でのみ取得できます。



次の図では、方向ごとに2つのキャプチャポイントが示されています (内部スイッチアーキテクチャによる)。



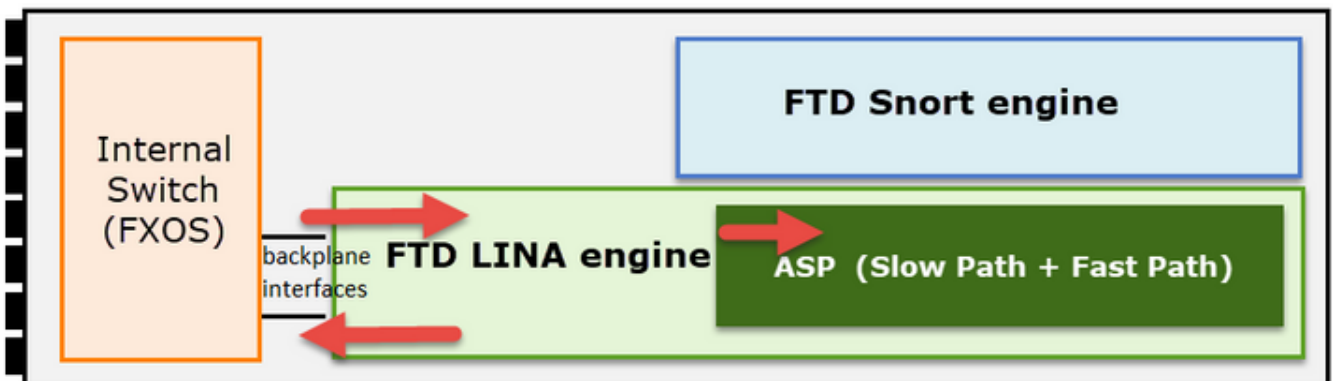
ポイント 2、3、および 4 でキャプチャされるパケットは、仮想ネットワークタグ (VNTag) を持ちます。

 注:FXOSシャーシレベルのキャプチャは、FP41xxおよびFP93xxプラットフォームでのみ使用できます。FP1xxx および FP21xx では、この機能は提供されません。

FTD LINA キャプチャの有効化および収集

主なキャプチャポイントは、次のとおりです。

- 入インターフェイス
- 出インターフェイス
- 高速セキュリティパス (ASP)



Firepower Management Center ユーザーインターフェイス (FMC UI) または FTD CLI のいずれかを使用して、FTD LINA キャプチャを有効にして収集することができます。

CLI から INSIDE インターフェイスでのキャプチャを有効にします。

```
<#root>
```

```
firepower#
```

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

このキャプチャは、IP 192.168.103.1 と 192.168.101.1 の間のトラフィックを双方向で照合します。

ASP キャプチャを有効にして、FTD LINA エンジンによってドロップされたすべてのパケットを表示します。

```
<#root>
```

```
firepower#
```

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

FTD LINA キャプチャを FTP サーバー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
```

FTD LINA キャプチャを TFTP サーバーにエクスポートします。

```
<#root>
```

```
firepower#
```

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

FMC バージョン 6.2.x 以降では、FMC UI から FTD LINA キャプチャを有効にして収集することができます。

FMC 管理対象ファイアウォールから FTD キャプチャを収集するもう一つの方法は、次のとおりです。

手順 1

LINAまたはASPキャプチャの場合は、キャプチャをFTDディスクにコピーします。

```
<#root>
```

```
firepower#
```

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

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

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

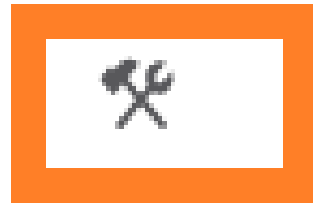
手順 2

エキスパートモードに移行し、保存されたキャプチャを findings に見つけて、それを /ngfw/var/common にコピーします。

```
<#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
```

手順 3

FTD を管理している FMC にログインし、[デバイス (Devices)] > [デバイス管理 (Device Management)] に移動します。FTD デバイスを見つけて、トラブルシューティングのアイコンを選択します。



手順 4

[高度なトラブルシューティング (Advanced Troubleshooting)] を選択します。

A screenshot of the Cisco Firepower Management Center (FMC) interface. The top navigation bar includes the Cisco logo, the text "Firepower Management Center", and a search icon. Below the navigation bar, the breadcrumb path is "System / Health / Health Monitor Appliance". The main heading is "Health Monitor". Below this, there is a table with one row for the appliance "mzafeiro_FP2110-2", which has a red warning icon. To the right of the appliance name are two buttons: "Generate Troubleshooting Files" and "Advanced Troubleshooting". The "Advanced Troubleshooting" button is highlighted with an orange border.

キャプチャファイル名を指定し、[ダウンロード (Download)] を選択します。

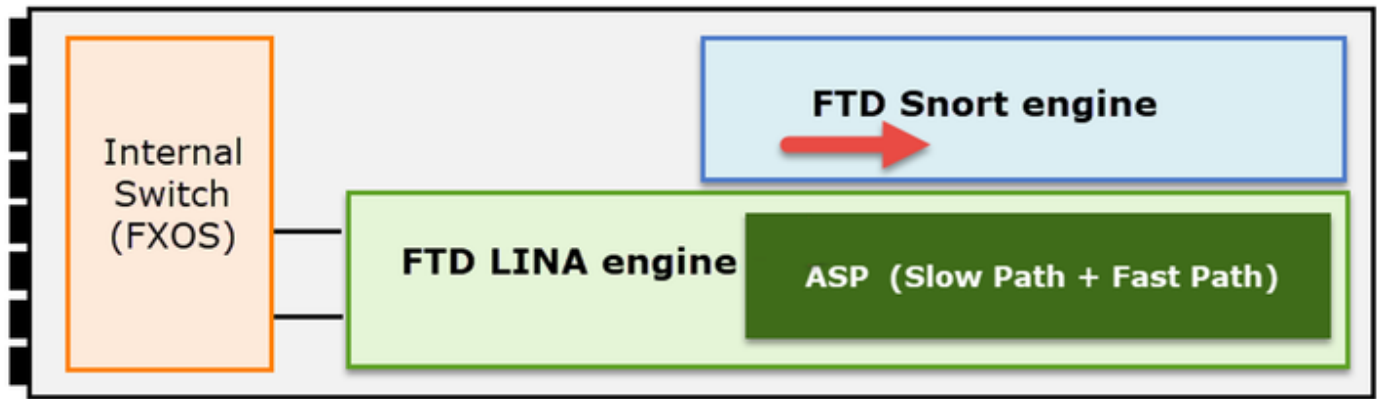
A screenshot of the Cisco Firepower Management Center (FMC) interface showing the "Advanced Troubleshooting" page. The breadcrumb path is "System / Health / AT File Download". The main heading is "Advanced Troubleshooting" for the appliance "mzafeiro_FP2110-2". Below the heading, there are four tabs: "File Download", "Threat Defense CLI", "Packet Tracer", and "Capture w/Trace". The "File Download" tab is selected. In the main content area, there is a "File" label above a text input field containing "capin.pcap". To the right of the input field are two buttons: "Back" and "Download". The "Download" button is highlighted with an orange border.

FMC UI からキャプチャを有効化/収集する方法の他の例については、次のドキュメントを参照してください。

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

FTD Snort キャプチャの有効化および収集

次の図にキャプチャポイントが示されています。



Snort レベルの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:
```

```
-n host 192.168.101.1
```

キャプチャを capture.pcap という名前のファイルに書き込み、FTP 経由でリモートサーバーにコピーするには、次の手順に従います。

```
<#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:

```
-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.

>

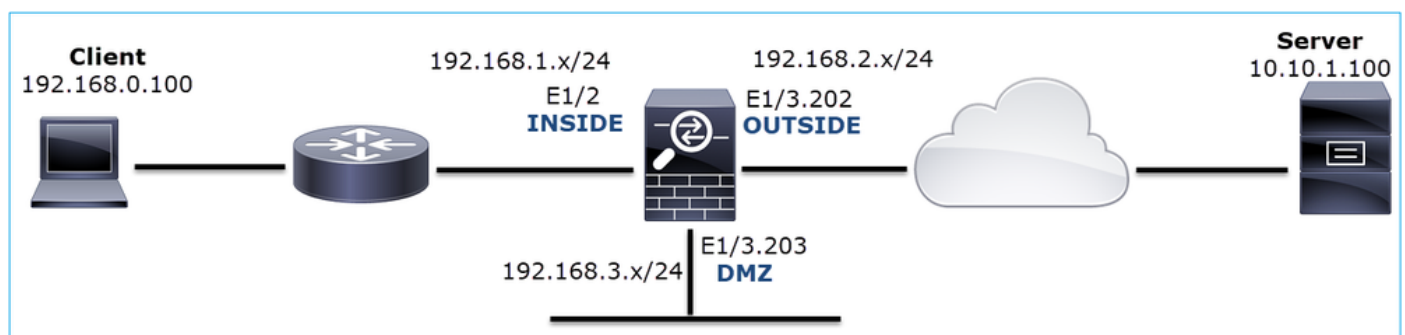
さまざまなキャプチャフィルタを含む Snort レベルのキャプチャの例については、次のドキュメントを参照してください。

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

トラブルシューティング

Case 1. 出カインターフェイスにTCP SYNなし

このトポロジは、次のとおりです。



問題の説明：HTTPが機能しない

影響を受けるフロー：

送信元IP:192.168.0.100

宛先IP:10.10.1.100

プロトコル : TCP 80

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

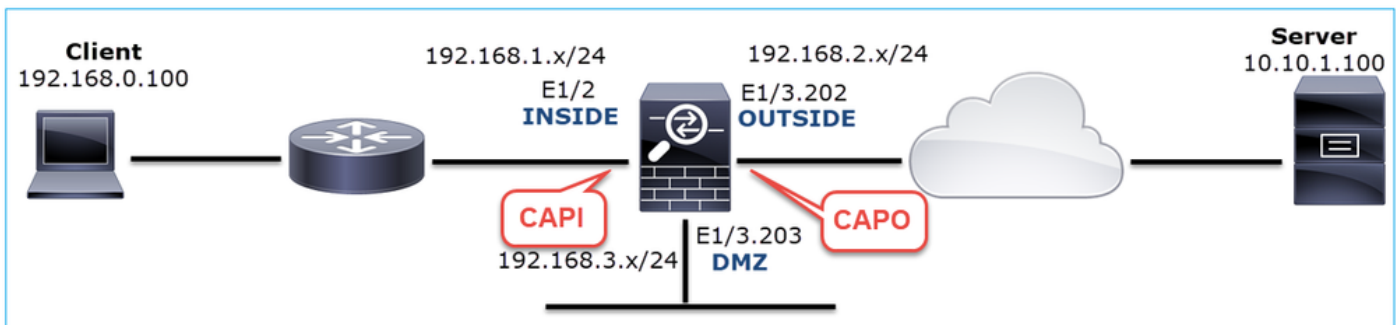
```
<#root>
```

```
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
```



キャプチャ - 機能シナリオ :

機能シナリオのキャプチャを持つておくことは、基準として常に非常に役立ちます。

次の図は、NGFW の INSIDE インターフェイスで取得されたキャプチャを示しています。

No.	Time	Source	Destination	Protocol	Length	Info
2	0.250878	192.168.0.100	10.10.1.100	TCP	66	1779 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	0.001221	10.10.1.100	192.168.0.100	TCP	66	80 → 1779 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
4	0.000488	192.168.0.100	10.10.1.100	TCP	54	1779 → 80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
5	0.000290	192.168.0.100	10.10.1.100	HTTP	369	GET / HTTP/1.1
6	0.002182	10.10.1.100	192.168.0.100	HTTP	966	HTTP/1.1 200 OK (text/html)
7	0.066830	192.168.0.100	10.10.1.100	HTTP	331	GET /welcome.png HTTP/1.1
8	0.021727	10.10.1.100	192.168.0.100	TCP	1434	80 → 1779 [ACK] Seq=913 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]
9	0.000000	10.10.1.100	192.168.0.100	TCP	1434	80 → 1779 [ACK] Seq=2293 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]
10	0.000626	192.168.0.100	10.10.1.100	TCP	54	1779 → 80 [ACK] Seq=593 Ack=3673 Win=66240 Len=0

> Frame 2: 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: 1779, Dst Port: 80, Seq: 0, Len: 0

キーポイント :

1. TCP 3ウェイハンドシェイクです。
2. 双方向のデータ交換です。
3. パケット間の遅延なし (パケット間の時間差に基づく)
4. 送信元 MAC は正しいダウンストリームデバイスです。

次の図は、NGFW の OUTSIDE インターフェイスで取得されたキャプチャを示しています。

No.	Time	Source	Destination	Protocol	Length	Info
2	0.250787	192.168.0.100	10.10.1.100	TCP	70	1779 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
3	0.000534	10.10.1.100	192.168.0.100	TCP	70	80 → 1779 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.000564	192.168.0.100	10.10.1.100	TCP	58	1779 → 80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
5	0.000534	192.168.0.100	10.10.1.100	HTTP	373	GET / HTTP/1.1
6	0.001663	10.10.1.100	192.168.0.100	HTTP	970	HTTP/1.1 200 OK (text/html)
7	0.067273	192.168.0.100	10.10.1.100	HTTP	335	GET /welcome.png HTTP/1.1
8	0.021422	10.10.1.100	192.168.0.100	TCP	1438	80 → 1779 [ACK] Seq=913 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]
9	0.000015	10.10.1.100	192.168.0.100	TCP	1438	80 → 1779 [ACK] Seq=2293 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]

> Frame 2: 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: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: 1779, Dst Port: 80, Seq: 0, Len: 0

キーポイント：

1. CAPI キャプチャと同じデータです。
2. 宛先 MAC は正しいアップストリームデバイスです。

キャプチャ - 非機能シナリオ：

デバイス CLI では、キャプチャは次のように示されます。

```
<#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 の内容：

```
<#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

次の図は、Wireshark での CAPI キャプチャを示しています。

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

キーポイント：

1. TCP SYN パケットのみが表示されています (TCP 3ウェイハンドシェイクは表示されていません)。

2. 確立できない 2 つの TCP セッション (送信元ポート 3171 および 3172) があります。送信元クライアントが TCP SYN パケットを再送信します。これらの再送信されたパケットは、Wireshark によって TCP 再送信として識別されます。
3. TCPの再送信は、~3秒から6秒の間隔で発生します。
4. 送信元 MAC アドレスは正しいダウンストリームデバイスのものであります。

2 つのキャプチャに基づいて、次のことが結論付けられます。

- 特定の 5 タプル (src/dst IP、src/dst ポート、プロトコル) のパケットが、予期されたインターフェイス (INSIDE) のファイアウォールに到着しています。
- パケットは、予期されたインターフェイス (OUTSIDE) でファイアウォールを出ていません。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:エミュレートされたパケットのトレースを確認します。

パケットトレースツールを使用して、ファイアウォールでのパケットの予期される処理方法を確認します。パケットがファイアウォール アクセス ポリシーによってドロップされた場合、エミュレートされたパケットのトレースの出力は、次のようなものになります。

<#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

アクション2:ライブパケットのトレースを確認します。

パケットトレースを有効にして、実際の TCP SYN パケットがファイアウォールによってどのように処理されたかを確認します。デフォルトでは、最初の 50 の入力パケットのみがトレースされます。

<#root>

firepower#

```
capture CAPI trace
```

キャプチャバッファをクリアします。

<#root>

firepower#

```
clear capture /all
```

パケットがファイアウォール アクセス ポリシーによってドロップされた場合、トレースの出力は、次のようなものになります。

<#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

アクション3:FTD Linaログを確認します。

FMC を介して FTD の Syslog を設定する方法については、次のドキュメントを参照してください。

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

FTD LINA ログ用に外部 Syslog サーバーを設定することを強くお勧めします。リモート Syslog サーバーが設定されていない場合は、トラブルシューティング中にファイアウォールでローカルバッファログを有効にします。この例に示されているログ設定は、設定を行うための優れた出発点となります。

```
<#root>
```

```
firepower#
```

```
show run logging
```

```
...
```

```
logging enable
```

```
logging timestamp
```

```
logging buffer-size 1000000
```

```
logging buffered informational
```

端末ページャを制御するには、端末ページャを 24 行に設定します。

```
<#root>
```

```
firepower#
```

```
terminal pager 24
```

キャプチャバッファをクリアします。

```
<#root>
```

```
firepower#
```

```
clear logging buffer
```

接続をテストし、パーサーフィルタを使用してログを確認します。この例では、パケットは、ファイアウォールアクセスポリシーによってドロップされています。

```
<#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
```

アクション4:ファイアウォールのASPドロップを確認します。

パケットがファイアウォールによってドロップされていると思われる場合は、ファイアウォールによってドロップされたすべてのパケットの数をソフトウェアレベルで確認できます。

```
<#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
```

キャプチャを有効にして、すべてのASPソフトウェアレベルのドロップを確認できます。

```
<#root>
```

```
firepower#
```

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

 ヒント：パケットの内容に興味がない場合は、パケットヘッダーだけをキャプチャできます（ヘッダーのみのオプション）。これにより、同じキャプチャバッファでより多くのパケットをキャプチャできます。さらに、キャプチャバッファのサイズ（デフォルトでは500KB）を最大32MBに増やすことができます（bufferオプション）。最後に、FTDバージョン6.3以降では、file-sizeオプションを使用して、最大10GBのキャプチャファイルを設定できます。その場合、キャプチャの内容はpcap形式でのみ表示されます。

キャプチャの内容を確認する場合、フィルタを使用して検索を絞り込むことができます。

```
<#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
```

この場合、パケットはすでにインターフェイスレベルでトレースされているため、ドロップの理由は ASP キャプチャに示されていません。パケットは 1 つの場所 (入力インターフェイスまたは ASP ドロップ) でしかトレースできないことに注意してください。その場合は、複数の ASP ドロップを取得し、特定の ASP ドロップ理由を設定することをお勧めします。推奨されるアプローチを次に示します。

1. 現在の ASP ドロップカウンタをクリアします。

```
<#root>
```

```
firepower#
```

```
clear asp drop
```

2. トラブルシューティングするフローを、ファイアウォールを介して送信します (テストを実行します) 。

3. ASPドロップカウンタをもう一度確認し、増加したカウンタをメモします。

```
<#root>
```

```
firepower#
```

```
show asp drop
```

```
Frame drop:
```

```
  No route to host (
```

```
no-route
```

```
)
```

```
  Flow is denied by configured rule (
```

```
acl-drop
```

```
)
```

4. 確認した特定のドロップの ASP キャプチャを有効にします。

```
<#root>
```

```
firepower#
```

```
capture ASP_NO_ROUTE type asp-drop no-route
```

```
firepower#
```

```
capture ASP_ACL_DROP type asp-drop acl-drop
```

5. トラブルシューティングするフローを、ファイアウォールを介して送信します (テストを実行します)。

6. ASP キャプチャを確認します。この場合、ルートがないためにパケットがドロップされています。

```
<#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
```

アクション5:FTD回線の接続テーブルをチェックします。

インターフェイス「X」から出ると予期されているパケットが、何らかの理由でインターフェイス「Y」から出る場合があります。ファイアウォールの出カインターフェイスは、次の動作順序に基づいて決定されます。

1. 確立された接続のルックアップ
2. ネットワークアドレス変換 (NAT) のルックアップ (UN-NAT (宛先 NAT) フェーズは、PBR およびルートのルックアップよりも優先されます)
3. ポリシーベース ルーティング (PBR)
4. ルーティングテーブルのルックアップ

FTD 接続テーブルを確認するには、次の手順に従います。

```
<#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

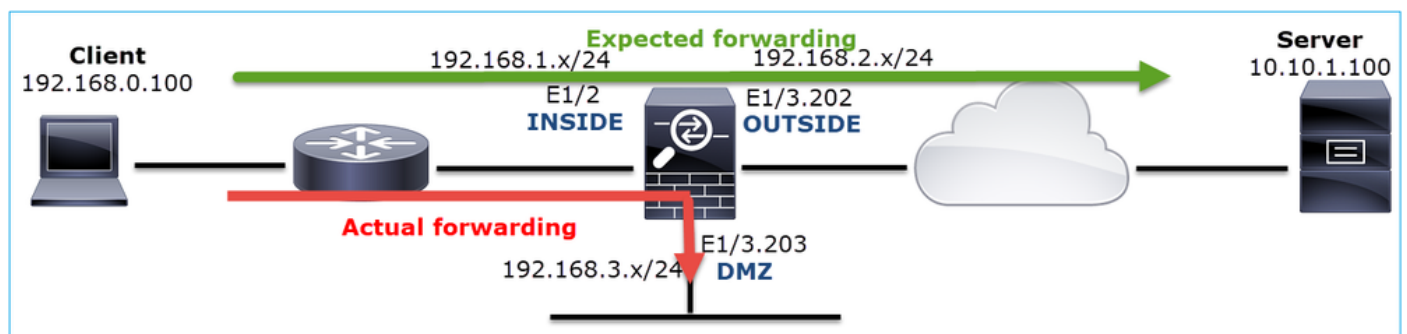
, idle 0:00:01, bytes 0, flags

aA N1


キーポイント：

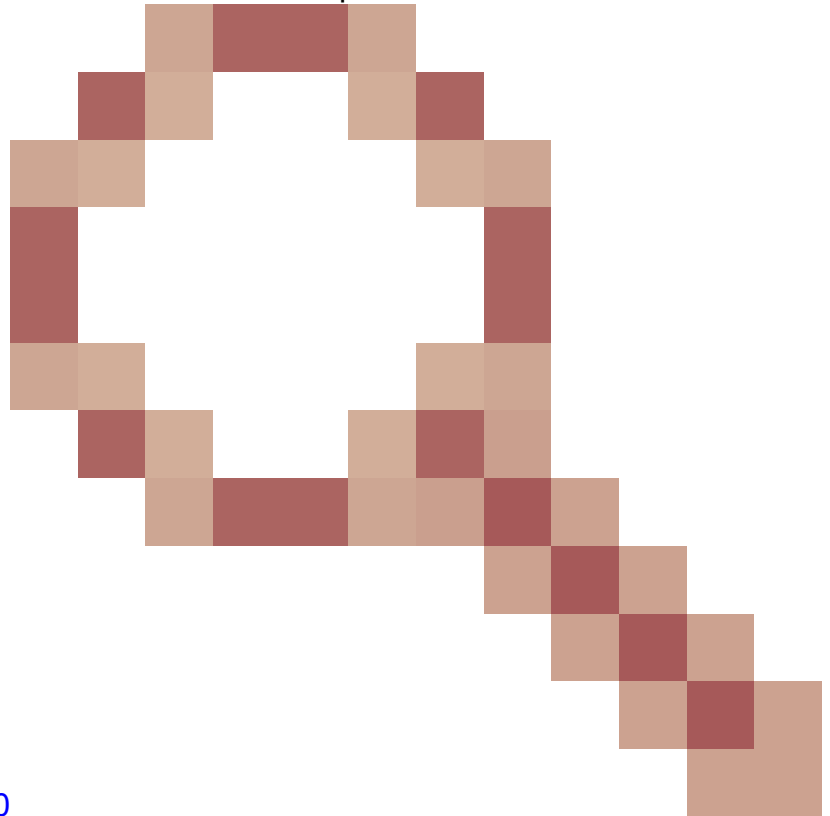
- フラグ (Aa) により、接続が初期状態 (ハーフオープン : TCP SYN のみがファイアウォールによって認識されている) であることが分かります。
- 送信元/宛先ポートにより、入カインターフェイスが INSIDE であり、出カインターフェイスが DMZ であることが分かります。

これは、次のように図示されます。



注：すべてのFTDインターフェイスのセキュリティレベルは0であるため、show conn の出力でのインターフェイスの順序はインターフェイス番号に基づきます。具体的には、vpif-

 num (仮想プラットフォームのインターフェイス番号) の大きいインターフェイスが内部として選択され、vpif-num の小さいインターフェイスが外部として選択されます。show interface detailコマンドを使用して、インターフェイスのvpif値を確認できます。関連する機



能拡張、Cisco Bug ID [CSCvi15290](#)

ENH:FTDはFTDの「show conn」出力に接続方向を示します

<#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

 注：Firepowerソフトウェアリリース6.5以降のASAリリース9.13.xでは、show conn longおよびshow conn detailコマンドの出力によって、接続の発信側と応答側に関する情報が提供されます

出力 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), fla
```

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

```
Connection lookup keyid: 228982375
```

出力 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
```

さらに、ネットワークアドレス変換の場合は、show conn long により、カッコ内に NAT 化された IP が表示されます。

```
<#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), fla
```

```
Initiator: 192.168.1.100, Responder: 192.168.2.222
```

```
Connection lookup keyid: 262895
```

アクション6:ファイアウォールのアドレス解決プロトコル(ARP)キャッシュをチェックします。

ファイアウォールがネクストホップを解決できない場合、そのファイアウォールは、元のパケット (この場合は TCP SYN) をサイレントにドロップし、ネクストホップを解決するまで ARP 要求を継続的に送信します。

ファイアウォール ARP キャッシュを表示するには、次のコマンドを使用します。

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

さらに、未解決のホストがあるかどうかを確認するには、次のコマンドを使用します。

```
<#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
```

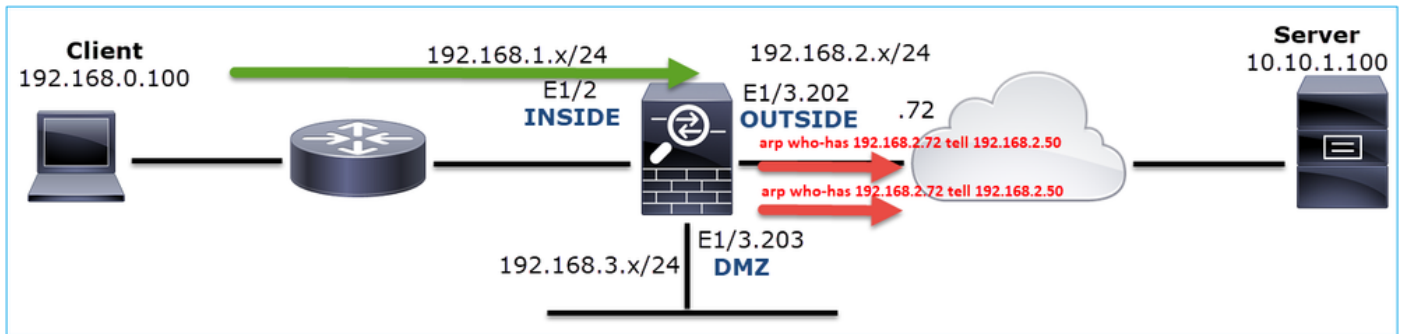
ARP の動作をさらに確認する場合は、ARP 固有のキャプチャを有効にすることができます。

```
<#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
```

この出力では、ファイアウォール (192.168.2.50) がネクストホップ (192.168.2.72) の解決を試みているが、ARP 応答はありません。



次の出力は、適切な ARP 解決が発生した機能シナリオを示しています。

```
<#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
```

適切な ARP エントリがない場合、ライブ TCP SYN パケットのトレースは次のようになります。

```
<#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

出力からわかるように、ネクストホップに到達できず、パケットがファイアウォールによって通知なしに廃棄された場合でも、トレースにはAction: allowと表示されます。この場合、より正確な出力が提供されるパケットトレーサツールも確認する必要があります。

<#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)

最近のASA/Firepowerバージョンでは、以前のメッセージは次の目的に最適化されています。

<#root>

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

Check ARP table (show arp) has entry for nexthop

., Drop-location: f

考えられる原因と推奨されるアクションの概要

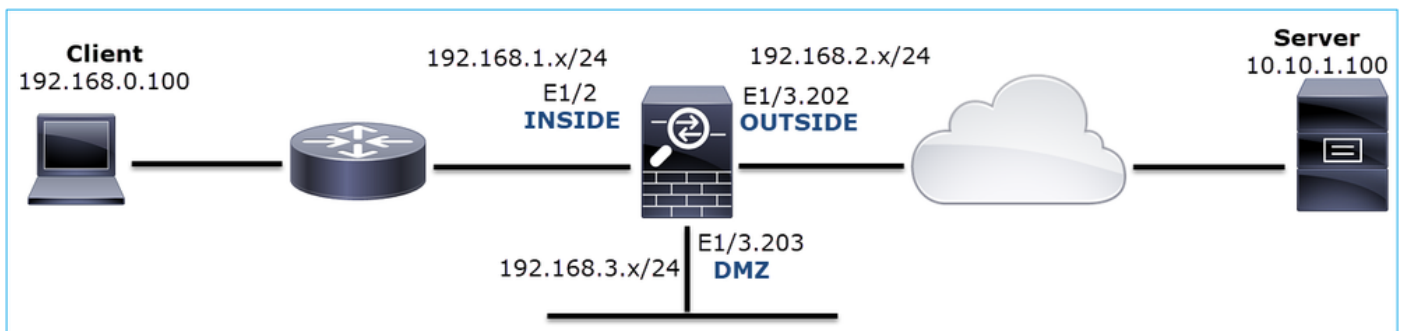
TCP SYN パケットが入カインターフェイスにしか表示されず、予期される出カインターフェイスから TCP SYN パケットが送信されない場合、考えられる原因は次のとおりです。

考えられる原因	推奨される対処法
パケットがファイアウォール アクセス ポリシーによってドロップされています。	<ul style="list-style-type: none">• packet-tracer または capture w/trace を使用して、ファイアウォールによるパケットの処理方法を確認します。• ファイアウォールログを確認します。• ファイアウォール ASP ドロップを確認します (show asp drop または capture type asp-drop)。• FMC 接続イベントを確認します。これは、ルールでログ記録が有効になっていることが前提となります。
キャプチャフィルタが適切ではありません。	<ul style="list-style-type: none">• packet-tracer または capture w/trace を使用して、送信元 IP または宛先 IP を変更する NAT 変換が存在するかどうかを確認します。この場合は、キャプチャフィルタを調整します。• show conn long コマンドの出力には、NAT 化された IP が示されます。
パケットが別の出カインターフェイスに送信されています。	<ul style="list-style-type: none">• packet-tracer または capture w/trace を使用して、ファイアウォールによるパケットの処理方法を確認します。出カインターフェイスの決定、現在の接続、UN-NAT、PBR、およびルーティングテーブルのルックアップに関する動作の順序を覚え

	<p>ておいてください。</p> <ul style="list-style-type: none"> ファイアウォールログを確認します。 ファイアウォール接続テーブルを確認します (show conn)。 <p>パケットが現在の接続に一致するために誤ったインターフェイスに送信される場合は、clear conn address コマンドを使用して、クリアする接続の5タプルを指定します。</p>
宛先へのルートが存在しません。	<ul style="list-style-type: none"> packet-tracer または capture w/trace を使用して、ファイアウォールによるパケットの処理方法を確認します。 ファイアウォール ASP ドロップ (show asp drop) で no-route ドロップの理由を確認します。
出カインターフェイスに ARP エントリがありません。	<ul style="list-style-type: none"> ファイアウォール ARP キャッシュを確認します (show arp)。 packet-tracer を使用して、有効な隣接関係 (アジャセンシー) が存在するかどうかを確認します。
出カインターフェイスがダウンしています。	<p>ファイアウォールに関する show interface ip brief コマンドの出力を調べて、インターフェイスのステータスを確認します。</p>

Case 2.クライアントからのTCP SYN、サーバからのTCP RST

次の図は、このトポロジを示しています。



問題の説明：HTTPが機能しない

影響を受けるフロー：

送信元IP:192.168.0.100

宛先IP:10.10.1.100

プロトコル：TCP 80

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

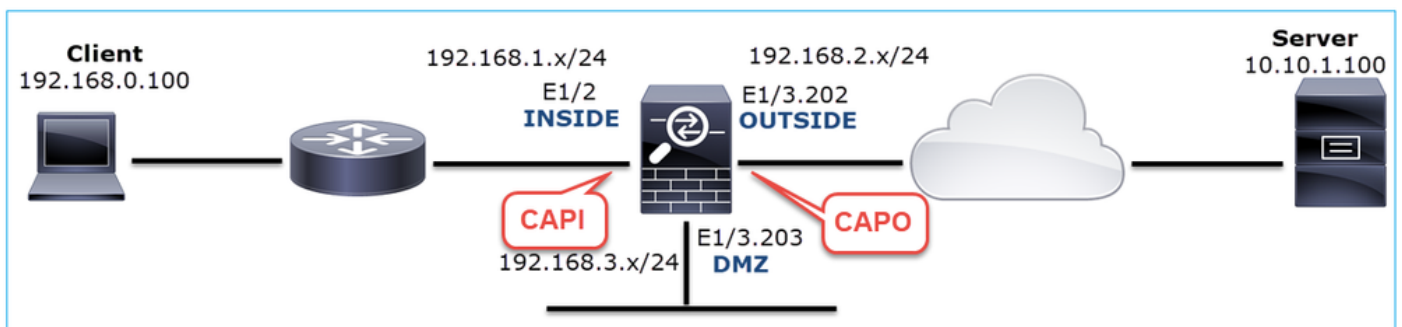
```
<#root>
```

```
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
```



キャプチャ - 非機能シナリオ：

デバイスCLIからのキャプチャは次のようになります。

```
<#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 の内容 :

```
<#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 の内容 :

```
<#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 PO 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>
```

次の図は、Wireshark での CAPI のキャプチャを示しています。

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: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

キーポイント：

1. 送信元が TCP SYN パケットを送信しています。
2. TCP RST が送信元に向けて送信されています。
3. 送信元が TCP SYN パケットを再送信しています。
4. MAC アドレスは適切です (入力パケットでは、送信元 MAC アドレスがダウンストリームルータに属し、宛先 MAC アドレスがファイアウォールの INSIDE インターフェイスに属しています) 。

次の図は、Wireshark での CAPO のキャプチャを示しています。

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) on interface 0
> Ethernet II, Src: Cisco_fc:fc:d8 (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: 22195, Dst Port: 80, Seq: 0, Len: 0

```

キーポイント：

1. 送信元が TCP SYN パケットを送信しています。
2. TCP RST が OUTSIDE インターフェイスに到着しています。
3. 送信元が TCP SYN パケットを再送信しています。
4. MAC アドレスは適切です (出力パケットでは、ファイアウォールの OUTSIDE が送信元 MAC であり、アップストリームルータが宛先 MAC です)。

2 つのキャプチャに基づいて、次のことが結論付けられます。

- クライアントとサーバー間の TCP 3ウェイハンドシェイクが完了していません。
- ファイアウォールの出カインターフェイスに到着する TCP RST が存在します。
- ファイアウォールは、MAC アドレスに基づいて適切なアップストリームおよびダウンストリームデバイスと「対話」しています。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:TCP RSTを送信する送信元MACアドレスをチェックします。

TCP SYN パケットに見られる宛先 MAC が、TCP RST パケットに見られる送信元 MAC と同じであることを確認します。

CAPO_RST_SERVER.pcap

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

Frame 2: 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: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: 22196, Dst Port: 80, Seq: 0, Len: 0

CAPO_RST_SERVER.pcap

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

Frame 3: 58 bytes on wire (464 bits) 58 bytes captured (464 bits)

Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8) Dst: Cisco_f6:1d:8e (00:be:75:f6:1d:8e)

802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202

Internet Protocol Version 4, Src: 10.10.1.100, Dst: 192.168.0.100

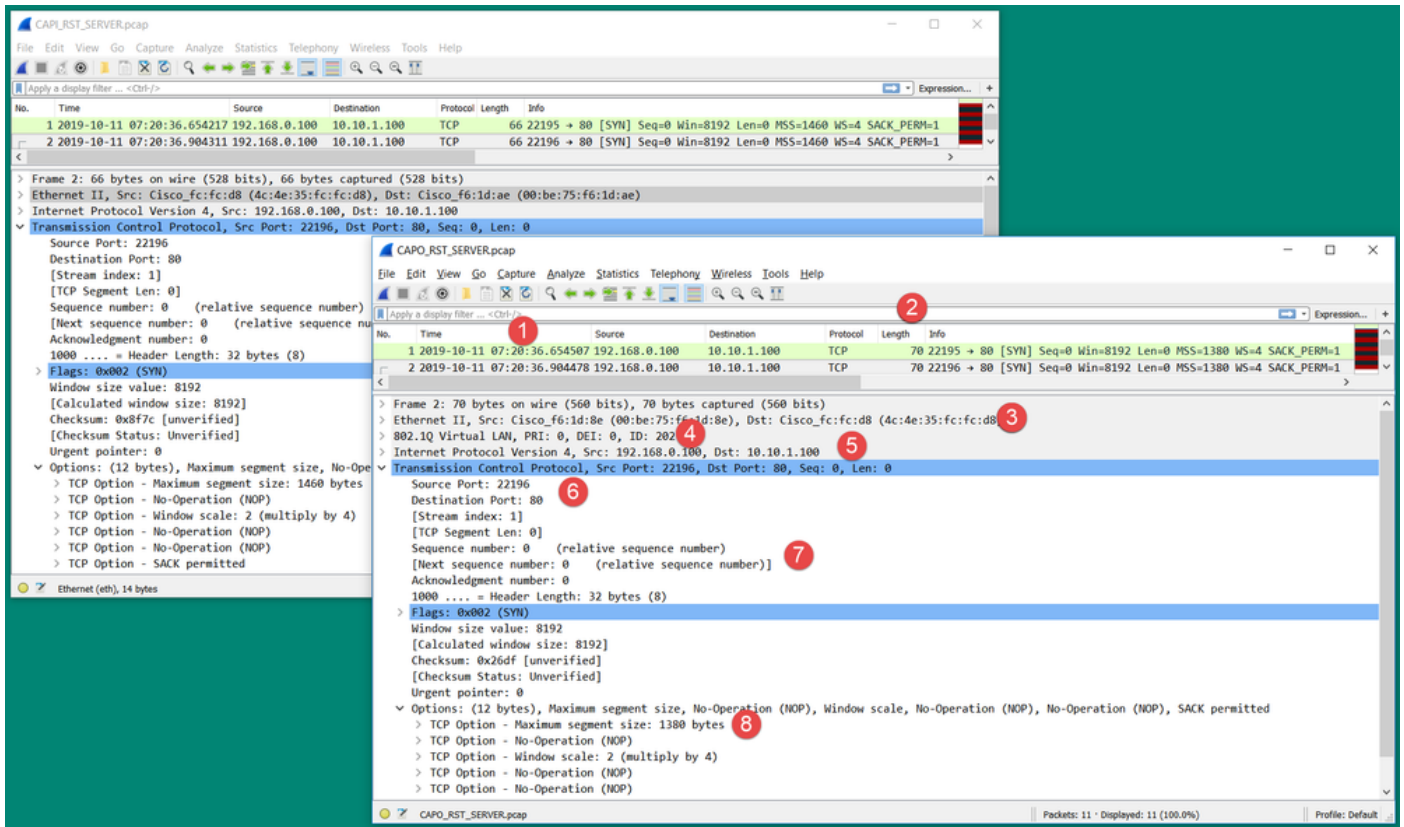
Transmission Control Protocol, Src Port: 80, Dst Port: 22196, Seq: 1, Ack: 1, Len: 0

この確認は、次の2つのことの確認が目的です。

- 非対称フローが存在しないことを確認します。
- MACが予期されるアップストリームデバイスに属していることを確認します。

アクション2:入力パケットと出力パケットを比較します。

Wireshark上の2つのパケットを視覚的に比較して、ファイアウォールがパケットを変更したり、破損させたりしていないことを確認します。いくつかの予期される相違点が強調表示されています。



キーポイント：

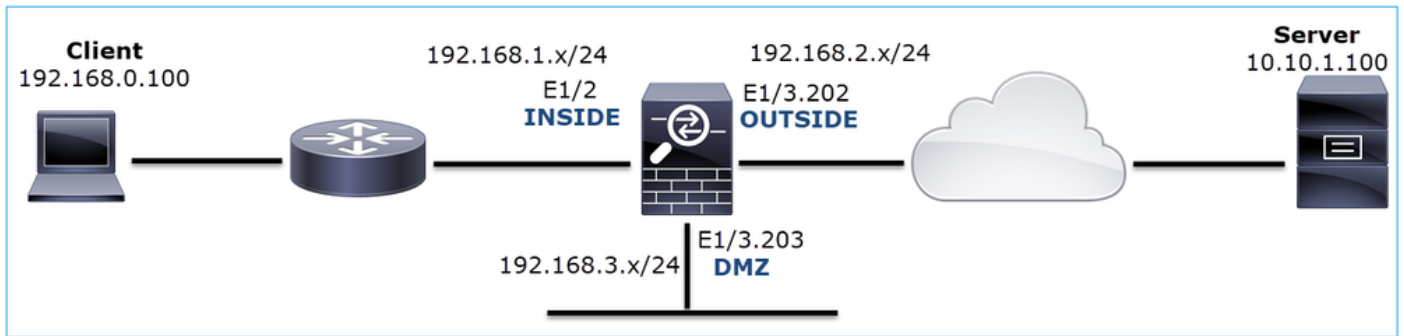
1. タイムスタンプが異なります。そのこと自体は問題ありませんが、その差は小さく、妥当な範囲である必要があります。これは、パケットに適用される機能およびポリシーチェックと、デバイスの負荷によって異なります。
2. パケットの長さは、特にdot1Qヘッダーが片側だけでファイアウォールによって追加/削除される場合に異なります。
3. MAC アドレスが異なります。
4. キャプチャがサブインターフェイスで取得された場合は、dot1Qヘッダーがある可能性があります。
5. NAT またはポートアドレス変換 (PAT) がパケットに適用されている場合、IP アドレスが異なります。
6. NAT または PAT がパケットに適用されている場合、送信元ポートまたは宛先ポートが異なります。
7. Wireshark の [相対シーケンス番号 (Relative Sequence Number)] オプションを無効にすると、初期シーケンス番号 (ISN) のランダム化により、ファイアウォールによって TCP シーケンス番号/確認応答番号が変更されていることが分かります。
8. 一部のTCPオプションは上書きできます。たとえば、ファイアウォールは、中継パスでのパケットのフラグメント化を避けるために、デフォルトで TCP 最大セグメントサイズ (MSS) を 1380 に変更します。

アクション3:目的地でキャプチャを取ります。

可能であれば、宛先自体でキャプチャを取得します。不可能な場合は、宛先のできるだけ近くでキャプチャを取得します。目的は、TCP RST の送信元 (宛先サーバーか、パス内の他のデバイスか) を確認することです。

Case 3.1つのエンドポイントからのTCP 3ウェイハンドシェイク+ RST

次の図は、このトポロジを示しています。



問題の説明：HTTPが機能しない

影響を受けるフロー：

送信元IP:192.168.0.100

宛先IP:10.10.1.100

プロトコル：TCP 80

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

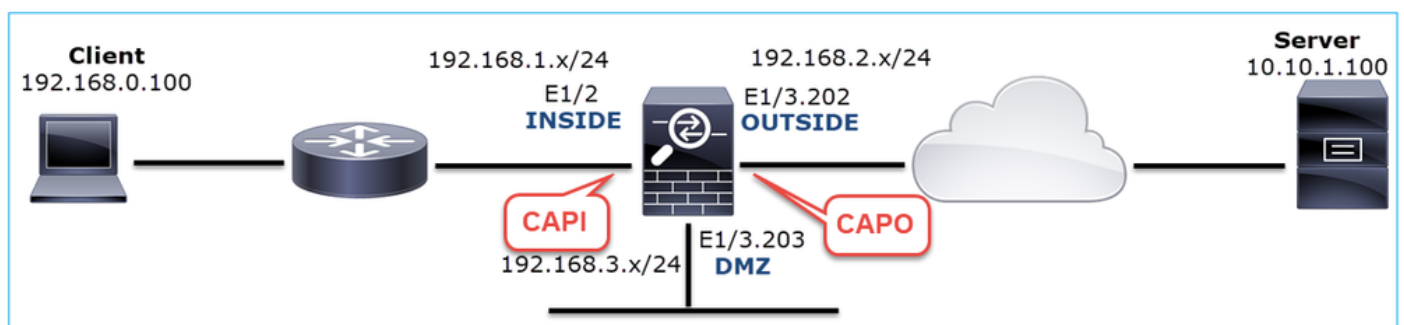
<#root>

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
```



キャプチャ - 非機能シナリオ :

この問題は、いくつかの異なる形でキャプチャに現れます。

3.1 : クライアントからの TCP 3 ウェイハンドシェイク + 遅延 RST

次の図のように、ファイアウォールキャプチャの CAPI と CAPO の両方に同じパケットが含まれています。

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

キーポイント :

1. TCP 3 ウェイハンドシェイクがファイアウォールを通過しています。
2. サーバーが SYN/ACK を再送信しています。
3. クライアントが ACK を再送信しています。
4. 約 20 秒後、クライアントは中断して TCP RST を送信しています。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:2つのエンドポイントにできるだけ近い場所からキャプチャを取得します。

ファイアウォールキャプチャは、クライアント ACK がサーバーによって処理されなかったことを示しています。これは、次の事実に基づいています。

- サーバーが SYN/ACK を再送信しています。
- クライアントが ACK を再送信しています。
- クライアントが、データの前に TCP RST または FIN/ACK を送信しています。

サーバーでのキャプチャは、問題の発生を示しています。TCP 3 ウェイハンドシェイクからのクライアント ACK が到着していません。

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 : クライアントからの TCP 3 ウェイハンドシェイク + 遅延 FIN/ACK + サーバーからの遅延 RST

次の図のように、ファイアウォールキャプチャの CAPI と CAPO の両方に同じパケットが含まれ

ています。

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

キーポイント：

1. TCP 3ウェイハンドシェイクがファイアウォールを通過しています。
2. 約 5 秒後、クライアントが FIN/ACK を送信しています。
3. 約 20 秒後、サーバーは中断して TCP RST を送信しています。

このキャプチャに基づいて、ファイアウォールを通過した TCP 3ウェイハンドシェイクが存在するものの1つのエンドポイントで実際に完了していないように見えると結論付けることができます（再送信がこれを示しています）。

推奨される対処法

ケース 3.1 と同じです。

3.3 : クライアントからの TCP 3ウェイハンドシェイク + 遅延 RST

次の図のように、ファイアウォールキャプチャの CAPI と CAPO の両方に同じパケットが含まれています。

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 WS=4 SACK_PERM=1
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	80 → 48355 [RST, ACK] Seq=2581697939 Ack=1633018699 Win=0 Len=0

キーポイント：

1. TCP 3ウェイハンドシェイクがファイアウォールを通過しています。
2. 約 20 秒後、クライアントは中断して TCP RST を送信しています。

これらのキャプチャに基づいて、次のことが結論付けられます。

- 5 ~ 20 秒後に、1つのエンドポイントが中断し、接続を終了することを決定しています。

推奨される対処法

ケース 3.1 と同じです。

3.4 : サーバーからの TCP 3ウェイハンドシェイク + 即時 RST

次の図のように、ファイアウォールキャプチャの CAPI と CAPO の両方にこれらのパケットが含

まれています。

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

キーポイント：

1. TCP 3ウェイハンドシェイクがファイアウォールを通過しています。
2. ACK パケットの数ミリ秒後に、サーバーからの TCP RST が存在しています。

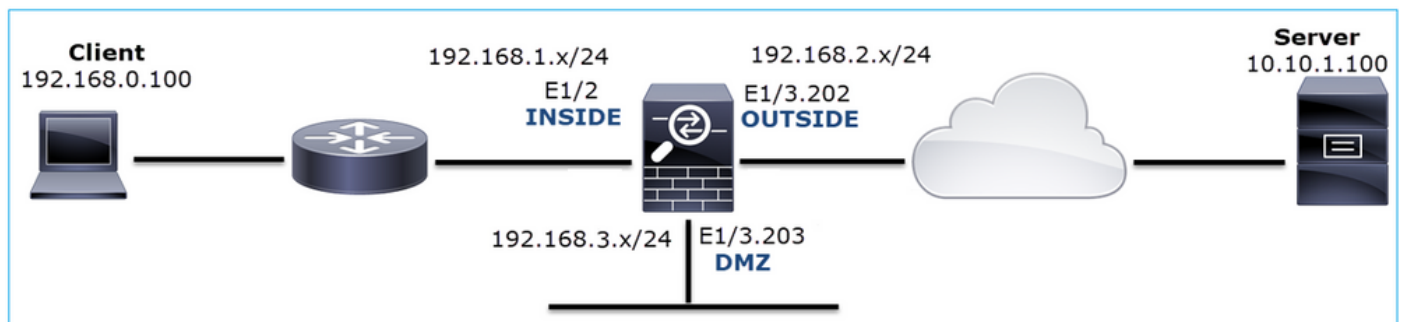
推奨される対処法

処置：可能な限りサーバの近くでキャプチャを実行します。

サーバーからの即時の TCP RST は、TCP RST を送信するパス内のサーバーまたはデバイスの誤動作を示している可能性があります。サーバー自体でキャプチャを取得し、TCP RST の送信元を特定します。

ケース 4.クライアントからのTCP RST

次の図は、このトポロジを示しています。



問題の説明：HTTP が機能しない

影響を受けるフロー：

送信元IP:192.168.0.100

宛先IP:10.10.1.100

プロトコル：TCP 80

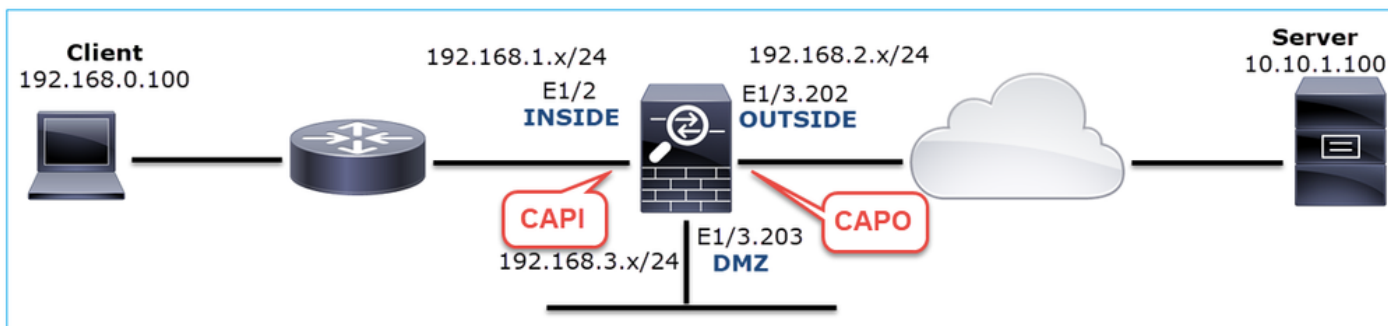
キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

<#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
```



キャプチャ - 非機能シナリオ :

CAPI の内容は、次のとおりです。

<#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

CAPO の内容は、次のとおりです。

<#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

ファイアウォールログは、次のようになります。

```
<#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
```

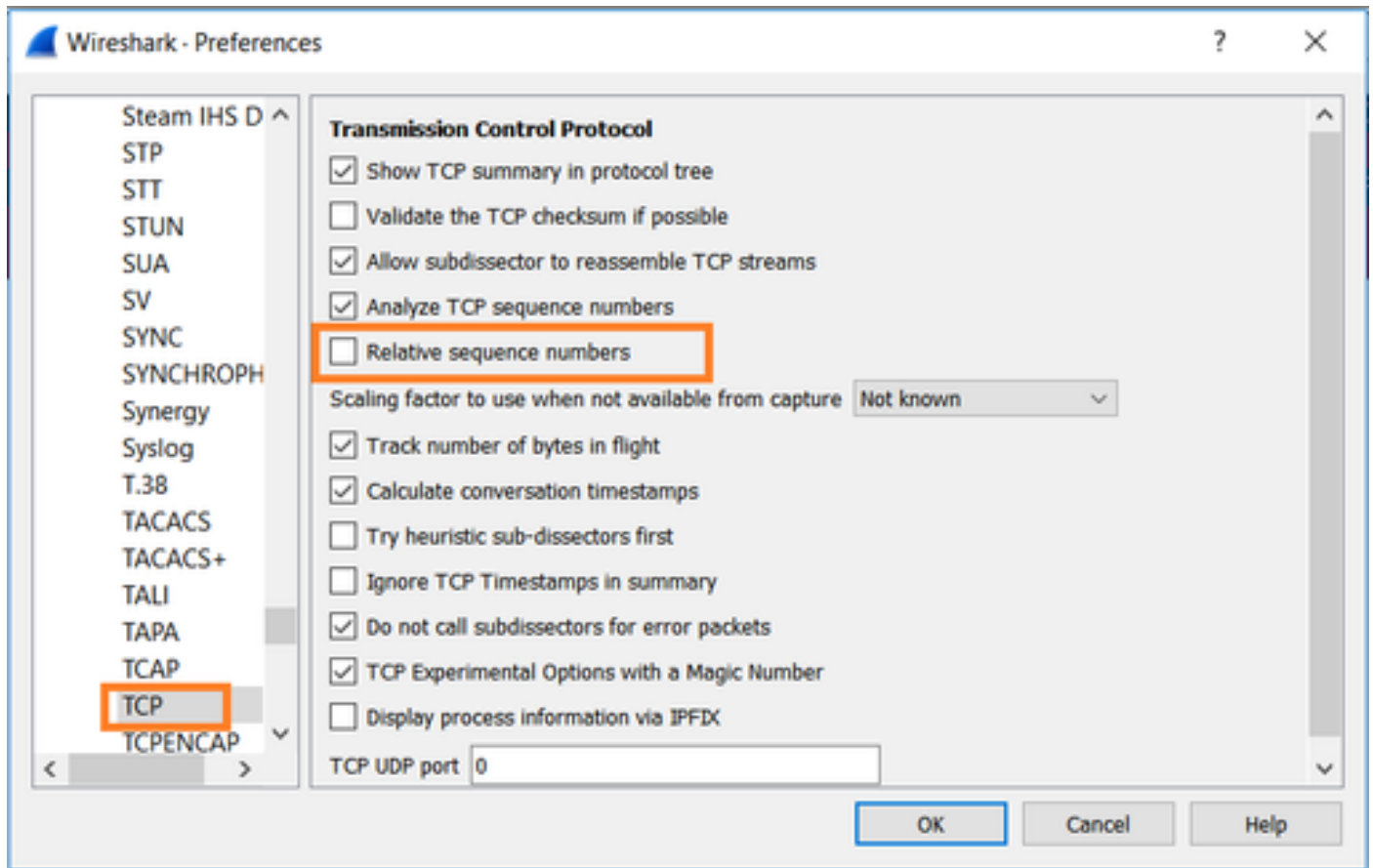
これらのログは、ファイアウォールの INSIDE インターフェイスに到着する TCP RST が存在することを示しています。

Wireshark での CAPI キャプチャ :

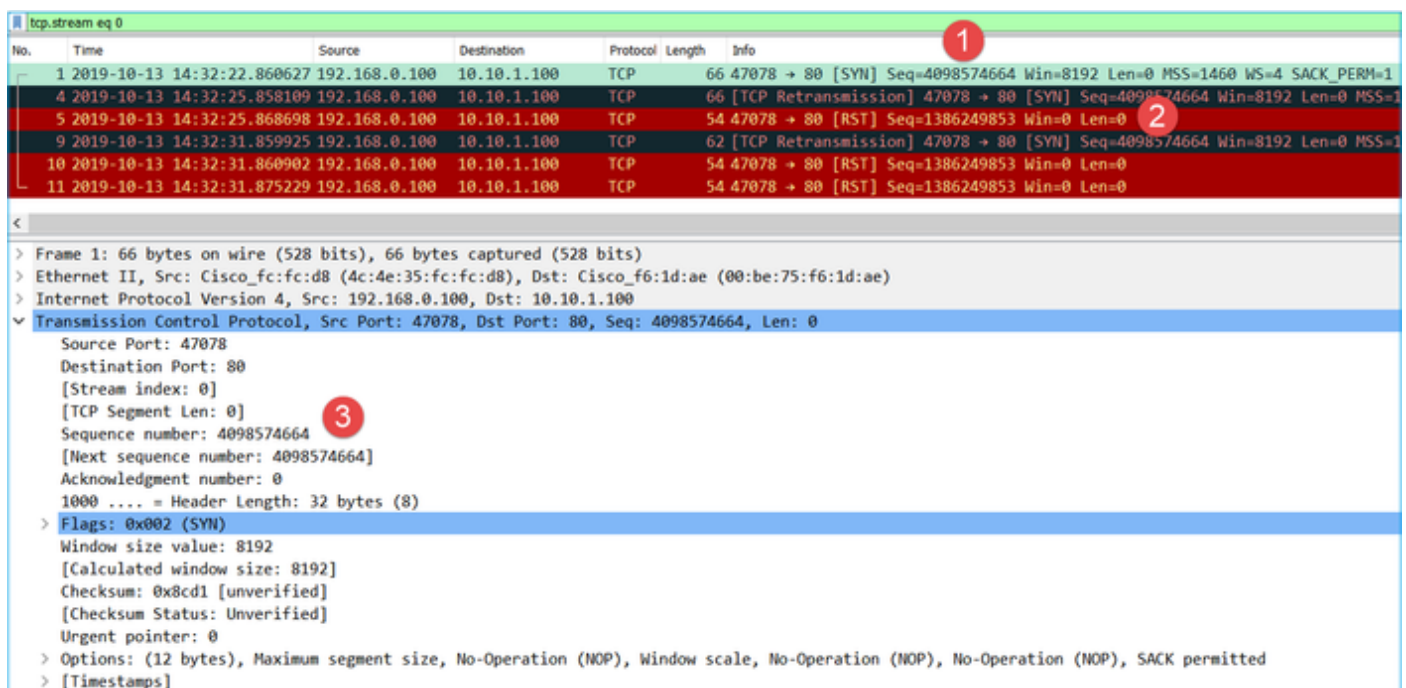
次の図のように、最初の TCP ストリームを追跡します。

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-13 14:32:22.860627	192.168.0.100	10.10.1.100	TCP	66	66 47078 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PE...
2	2019-10-13 14:32:23.111307	192.168.0.100	10.10.1.100	TCP	66	66 47079 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PE...
3	2019-10-13 14:32:23.112390	192.168.0.100	10.10.1.100	TCP	54	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0
4	2019-10-13 14:32:25.858109	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 47078 → 80 [SYN] Seq=0 Win=8192 Len=0
5	2019-10-13 14:32:25.868698	192.168.0.100	10.10.1.100	TCP	54	54 47078 → 80 [RST] Seq=1582642485 Win=0 Len=0
6	2019-10-13 14:32:26.108118	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 47079 → 80 [SYN] Seq=0 Win=8192 Len=0
7	2019-10-13 14:32:26.109079	192.168.0.100	10.10.1.100	TCP	54	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0
8	2019-10-13 14:32:26.118295	192.168.0.100	10.10.1.100	TCP	54	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0
9	2019-10-13 14:32:31.859925	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 47078 → 80 [SYN] Seq=0 Win=8192 Len=0
10	2019-10-13 14:32:31.860902	192.168.0.100	10.10.1.100	TCP	54	54 47078 → 80 [RST] Seq=1582642485 Win=0 Len=0
11	2019-10-13 14:32:31.875229	192.168.0.100	10.10.1.100	TCP	54	54 47078 → 80 [RST] Seq=1582642485 Win=0 Len=0
12	2019-10-13 14:32:32.140632	192.168.0.100	10.10.1.100	TCP	54	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0
13	2019-10-13 14:32:32.159995	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 47079 → 80 [SYN] Seq=0 Win=8192 Len=0
14	2019-10-13 14:32:32.160956	192.168.0.100	10.10.1.100	TCP	54	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0

[Wireshark] で、[編集 (Edit)] > [設定 (Preferences)] > [プロトコル (Protocols)] > [TCP] に移動し、図のように、[相対シーケンス番号 (Relative sequence numbers)] オプションをオフにします。



次の図は、CAPI キャプチャにおける最初のフローの内容を示しています。



キーポイント：

1. クライアントが TCP SYN パケットを送信しています。
2. クライアントが TCP RST パケットを送信しています。
3. TCP SYN パケットのシーケンス番号の値は 4098574664 です。

CAPO キャプチャの同じフローには、次のものが含まれます。

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

キーポイント：

1. クライアントが TCP SYN パケットを送信しています。ファイアウォールが ISN をランダム化しています。
2. クライアントが TCP RST パケットを送信しています。

2つのキャプチャに基づいて、次のことが結論付けられます。

- ・ クライアントとサーバーの間に TCP 3 ウェイハンドシェイクは存在しません。
- ・ クライアントから送信された TCP RST が存在します。CAPI キャプチャの TCP RST シーケンス番号の値は 1386249853 です。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:クライアントでキャプチャを取得します。

ファイアウォールで収集されたキャプチャによると、非対称フローの強い兆候が存在します。これは、クライアントが 1386249853の値 (ランダム化された ISN) で TCP RST を送信しているという事実に基づいています。

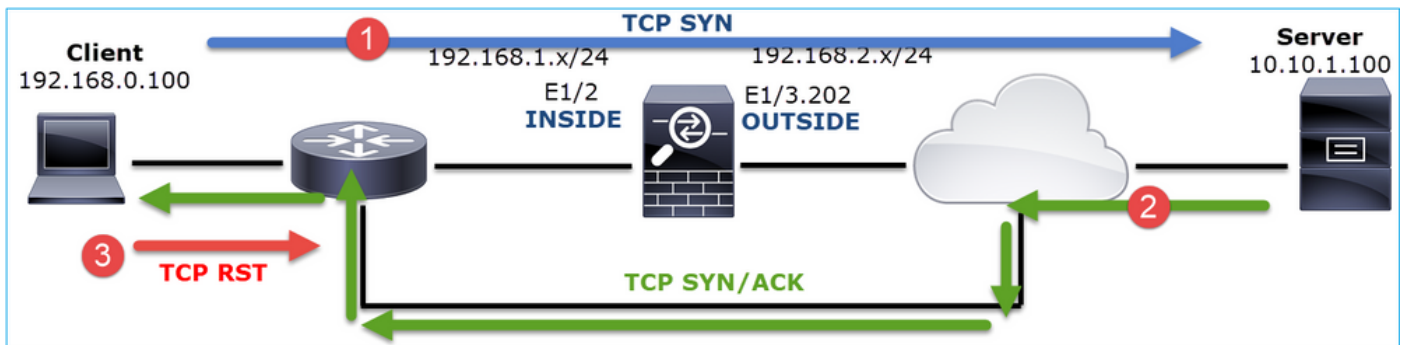
No.	Time	Source	Destination	Protocol	Length	Info
19	6.040337	192.168.0.100	10.10.1.100	TCP	66	47078→80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
29	9.037499	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 47078→80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1460 WS=
30	9.048155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

キーポイント：

1. クライアントが TCP SYN パケットを送信しています。シーケンス番号は 4098574664 であり、ファイアウォールの INSIDE インターフェイス (CAPI) で見られるものと同じです。
2. ACK 番号が 1386249853 の TCP SYN/ACK が存在します (これは ISN のランダム化によって予期されるものです)。このパケットは、ファイアウォールキャプチャには表示されていません。

3. クライアントは、ACK 番号の値が 4098574665 の SYN/ACK を予期していましたが、受信した値は 1386249853 であったため、TCP RST を送信しています。

それを図で示します。

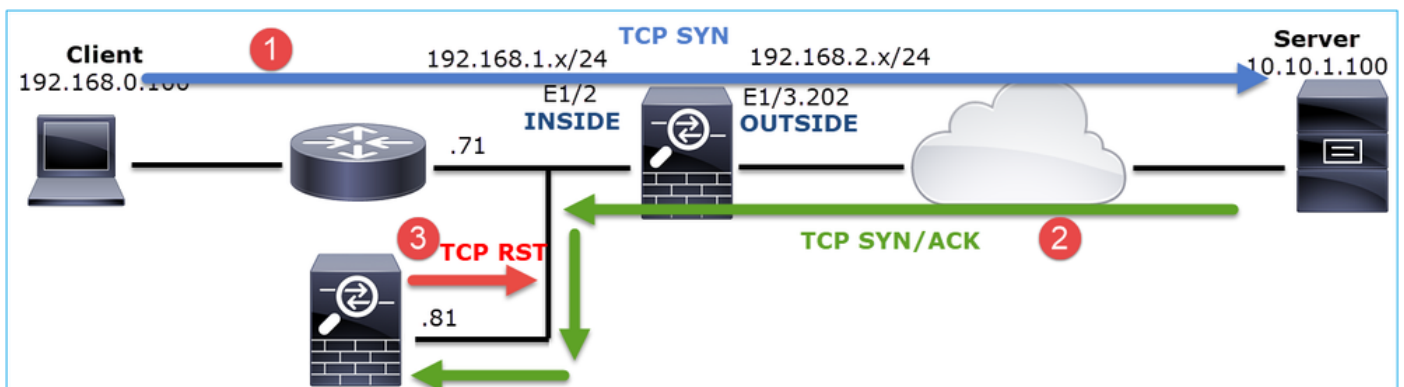


アクション2:クライアントとファイアウォール間のルーティングを確認します。

次の項目を確認します。

- キャプチャに表示される MAC アドレスが、予期されたものである。
- ファイアウォールとクライアントの間のルーティングが対称である。

内部ネットワークに非対称ルーティングが存在するときにファイアウォールとクライアントの間にあるデバイスから RST が送信されるシナリオがあります。次の図は、その典型的なケースを示しています。



この場合、キャプチャの内容は、次のようになります。TCP SYN パケットの送信元 MAC アドレスと TCP RST の送信元 MAC アドレスおよび TCP SYN/ACK パケットの宛先 MAC アドレスの違いに注意してください。

```
<#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)
```

...

ケース 5.遅いTCP転送 (シナリオ1)

事象の説明:

ホスト 10.11.4.171 とホスト 10.77.19.11 の間の SFTP 転送が低速になっています。2つのホスト間の最小帯域幅 (BW) は 100 Mbps ですが、転送速度は 5 Mbps を超えていません。

一方で、ホスト 10.11.2.124 とホスト 172.25.18.134 の間の転送速度はかなり高速です。

背景理論:

単一の TCP フローの最大転送速度は、帯域幅遅延積 (BDP) によって決定されます。次の図は、使用される式を示しています。

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

BDP の詳細については、次の資料を参照してください。

- [『Why Your Application only Uses 10Mbps Even the Link is 1Gbps?』](#)
- [BRKSEC-3021 - Advanced : 『Maximizing Firewall Performance』](#)

シナリオ 1.遅い転送

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:10.11.4.171

宛先IP:10.77.19.11

プロトコル：SFTP(FTP over SSH)

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

```
<#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
```

警告:FP1xxxおよびFP21xxのキャプチャLINAキャプチャは、FTDを通過するトラフィックの転送速度に影響を与えます。パフォーマンスの問題（FTDを介した低速転送）をトラブルシューティングする場合は、FP1xxx および FP21xxx プラットフォームで LINA キャプチャを有効にしないでください。代わりに、送信元ホストおよび宛先ホストでのキャプチャに加えて、SPAN または HW タップデバイスを使用してください。この問題は、Cisco Bug ID [CSCvo30697](https://cisco.com/bug/CSCvo30697)に記載されています。

```
<#root>
```

```
firepower#
```

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

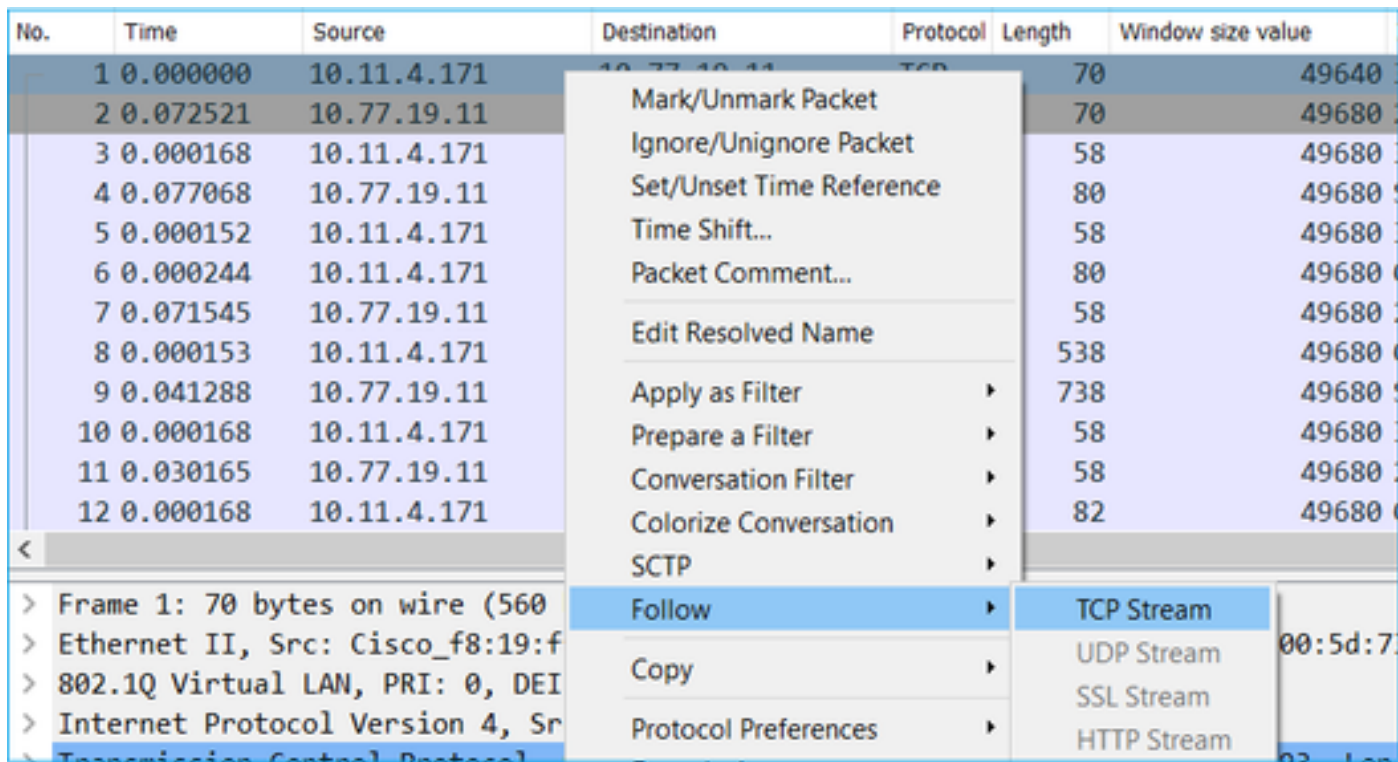
```
WARNING: Running packet capture can have an adverse impact on performance.
```

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

ラウンドトリップ時間 (RTT) 計算

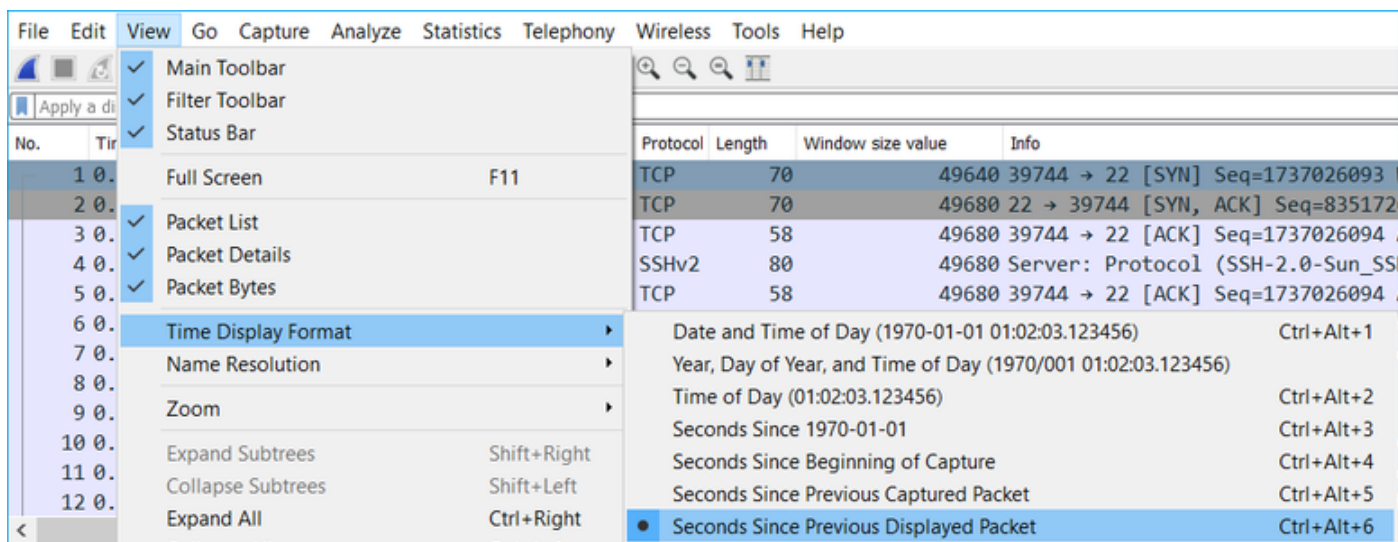
まず、転送フローを特定し、それを追跡します。



No.	Time	Source	Destination	Protocol	Length	Window size value
1	0.000000	10.11.4.171	10.77.19.11	TCP	70	49640
2	0.072521	10.77.19.11	10.11.4.171	TCP	70	49680
3	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680
4	0.077068	10.77.19.11	10.11.4.171	TCP	80	49680
5	0.000152	10.11.4.171	10.77.19.11	TCP	58	49680
6	0.000244	10.11.4.171	10.77.19.11	TCP	80	49680
7	0.071545	10.77.19.11	10.11.4.171	TCP	58	49680
8	0.000153	10.11.4.171	10.77.19.11	TCP	538	49680
9	0.041288	10.77.19.11	10.11.4.171	TCP	738	49680
10	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680
11	0.030165	10.77.19.11	10.11.4.171	TCP	58	49680
12	0.000168	10.11.4.171	10.77.19.11	TCP	82	49680

Frame 1: 70 bytes on wire (560 bytes captured)	0:5d:7e:83:1a:03 (en0)
Ethernet II, Src: Cisco_f8:19:f0:0d:5d:7e, Dst: 08:00:27:00:00:00	
802.1Q Virtual LAN, PRI: 0, DEI: 0, Len: 54	
Internet Protocol Version 4, Src: 10.11.4.171, Dst: 10.77.19.11	
Transmission Control Protocol, Seq=1737026093, Len=22	

Wireshark の [表示 (View)] を変更して、[前に表示されたパケットからの秒数 (Seconds Since the Previous Displayed Packet)] を表示します。これにより、RTT の計算が容易になります。



File	Edit	View	Go	Capture	Analyze	Statistics	Telephony	Wireless	Tools	Help
<input checked="" type="checkbox"/> Main Toolbar										
<input checked="" type="checkbox"/> Filter Toolbar										
<input checked="" type="checkbox"/> Status Bar										
<input type="checkbox"/> Full Screen										
<input checked="" type="checkbox"/> Packet List										
<input checked="" type="checkbox"/> Packet Details										
<input checked="" type="checkbox"/> Packet Bytes										
Time Display Format										
Name Resolution										
Zoom										
Expand Subtrees										
Collapse Subtrees										
Expand All										

Protocol	Length	Window size value	Info
TCP	70	49640 39744 → 22	[SYN] Seq=1737026093
TCP	70	49680 22 → 39744	[SYN, ACK] Seq=835172
TCP	58	49680 39744 → 22	[ACK] Seq=1737026094
SSHv2	80		49680 Server: Protocol (SSH-2.0-Sun_SSH)
TCP	58	49680 39744 → 22	[ACK] Seq=1737026094

RTTは、2つのパケット交換 (1つは送信元に向かうパケット交換で、もう1つは宛先に向かうパケット交換) の間の時間値の加算によって計算できます。今回の場合、パケット番号 2 は、ファイアウォールと SYN/ACK パケットを送信したデバイス (サーバー) の間の RTT を示しています。パケット番号 3 は、ファイアウォールと ACK パケットを送信したデバイス (クライアント) の間の RTT を示しています。2つの数値を加算すると、エンドツーエンドの RTT に関する適

切な概算値が得られます。

1	0.000000	10.11.4.171	10.77.19.11	TCP	70	49680 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 ミリ秒

TCP ウィンドウサイズの計算

TCP パケットを展開して、TCP ヘッダーを展開し、[計算されたウィンドウサイズ (Calculated window size)] を選択して、[列として適用 (Apply as Column)] を選択します。

▼ 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]
Unsent sequence: 0

The scaled window size (if scaling has been ...)

Expand Subtrees
Collapse Subtrees
Expand All
Collapse All
Apply as Column

[計算されたウィンドウサイズ (Calculated window size)] 列の値を調べて、TCP セッション中の最大ウィンドウサイズ値を確認します。列名を選択して値をソートすることも可能です。

ファイルのダウンロード (サーバーからクライアントへ) をテストする場合は、サーバーによってアダバタイズされる値を確認する必要があります。サーバーによってアダバタイズされる最大ウィンドウサイズの値によって、達成される最大転送速度が決まります。

この場合、TCP ウィンドウサイズは約 50000 バイトです。

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)

これらの値に基づき、帯域幅遅延積の式を使用すると、理論上の最大帯域幅 $50000 * 8 / 0.08 = 5$

Mbpsという条件下で達成可能な理論上の最大帯域幅が得られます。

これは、今回のクライアントの状況と一致します。

TCP 3ウェイハンドシェイクを詳しく確認します。両側（特に重要なのはサーバー）は、 $2^0 = 1$ （ウィンドウスケールなし）を意味する0のウィンドウスケール値をアドバタイズしています。これは、転送速度に悪影響を与えます。

```
No.    Time    Source          Destination      Protocol Length  Window size value  Info
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

<
> Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
> Ethernet II, Src: Cisco_1f:72:4e (00:5d:73:1f:72:4e), Dst: Cisco_f8:19:ff (00:22:bd:f8:19:ff)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
> Internet Protocol Version 4, Src: 10.77.19.11, Dst: 10.11.4.171
> Transmission Control Protocol, Src Port: 22, Dst Port: 39744, Seq: 835172681, Ack: 1737026094, Len: 0
  Source Port: 22
  Destination Port: 39744
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 835172681
  [Next sequence number: 835172681]
  Acknowledgment number: 1737026094
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x012 (SYN, ACK)
  Window size value: 49680
  [Calculated window size: 49680]
  Checksum: 0xa91b [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
    > TCP Option - Maximum segment size: 1380 bytes
    > TCP Option - No-Operation (NOP)
    > TCP Option - Window scale: 0 (multiply by 1)
    > TCP Option - No-Operation (NOP)
```

この時点で、サーバ上でキャプチャを実行し、ウィンドウスケール=0をアドバタイズしたサーバであることを確認し、再設定する必要があります（この方法については、サーバのマニュアルを参照してください）。

シナリオ 2.高速転送

次に、優れたシナリオ（同じネットワークを介した高速転送）について説明します。

トポロジ：



関連するフロー：

送信元IP:10.11.2.124

宛先IP:172.25.18.134

プロトコル：SFTP(FTP over SSH)

FTD LINA エンジンでのキャプチャを有効にします。

<#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
```

ラウンドトリップ時間(RTT)の計算：この場合、RTTは約 300ミリ秒です。

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ウィンドウサイズの計算：サーバはTCPウィンドウスケール係数7をアドバタイズします。

```
> Internet Protocol Version 4, Src: 172.25.18.134, Dst: 10.11.2.124
v 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
  v 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]
```

サーバーの TCP ウィンドウサイズは約 1600000 バイトです。

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=

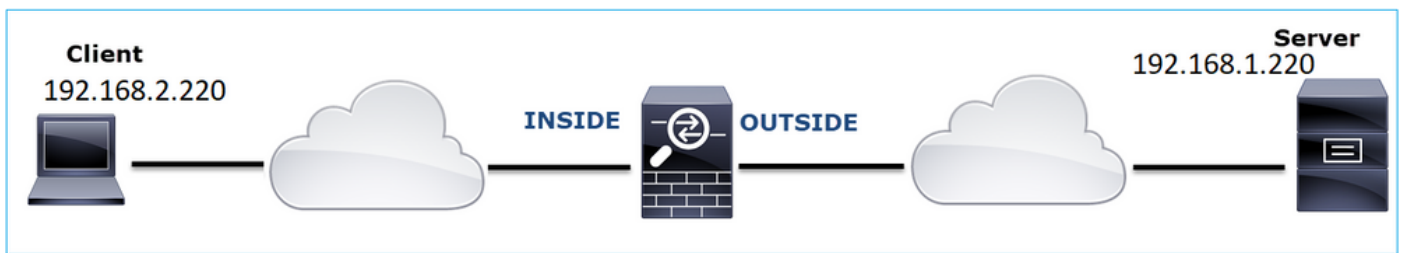
これらの値に基づき、帯域幅遅延積の式は次のようになります。

$$1600000 * 8 / 0.3 = 43 \text{ Mbps の最大理論転送速度}$$

Case 6.遅いTCP転送 (シナリオ2)

問題の説明：ファイアウォール経由のFTPファイル転送 (ダウンロード) が遅い。

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:192.168.2.220

宛先IP:192.168.1.220

プロトコル：FTP

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

```
<#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 パケットを選択し、FTD INSIDE キャプチャ (CAPI) の FTP データチャンネルを追跡

します。

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	Mark/Unmark Packet (PASV) (RETR file15mb)
78	0.000046	192.168.1.220	192.168.2.220	FTP-DATA	Ignore/Unignore Packet not captured] FTP Data: 124
79	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	Set/Unset Time Reference (PASV) (RETR file15mb)
80	0.000107	192.168.2.220	192.168.1.220	TCP	Time Shift... Seq=1884231612 Ack=2670019631
81	0.000092	192.168.2.220	192.168.1.220	TCP	Packet Comment... Seq=1884231612 Ack=2670020879
82	0.000091	192.168.2.220	192.168.1.220	TCP	Edit Resolved Name 4494 → 2388 [ACK] Seq=188423
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	(PASV) (RETR file15mb)
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	Conversation Filter 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	Colorize Conversation 88 → 54494 [ACK] Seq=2670020
89	0.000397	192.168.2.220	192.168.1.220	TCP	SCTP 2670027119
90	0.000869	192.168.1.220	192.168.2.220	FTP-DATA	Follow TCP Stream file15mb)

FTP-DATA ストリームの内容：


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=3577288500 TSecr=0 WS=128
28	1.026534	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	1.981584	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2669998978 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=2669998979 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
34	0.001617	192.168.2.220	192.168.1.220	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=26699990927 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
36	0.000458	192.168.1.220	192.168.2.220	FTP-DATA	1314 [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=26699990927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699992175 SRE=2669993423
39	0.000077	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=26699990927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699992175 SRE=2669994671
40	0.000056	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=26699990927 Ack=1884231612 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
41	0.000458	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	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
44	0.000077	192.168.2.220	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	0.000030	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	0.000054	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	1314 [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	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 [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=2670007151 SRE=2670007151
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=2670004655 SRE=2670008399
60	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

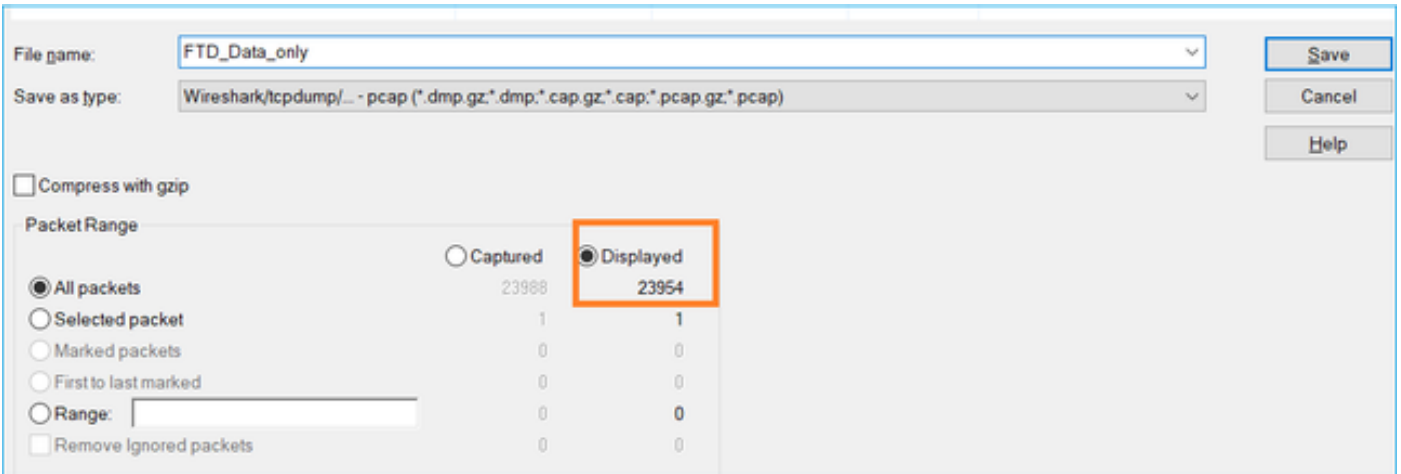
CAPO キャプチャの内容：

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=2157030681 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	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
42	0.000045	192.168.2.220	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=2224319408 SRE=2224321904
45	0.309005	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=3577291511
46	0.000580	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224319084 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
47	0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
48	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 [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	192.168.2.220	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=2224326400 SRE=2224328144
54	0.918019	192.168.1.220	192.168.2.220	TCP	1314 [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	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 [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)

キーポイント：

1. TCP Out-Of-Order (OOO) パケットが存在します。
2. TCP 再送信が存在します。
3. パケット損失 (ドロップされたパケット) の兆候が存在します。

 ヒント:File > Export Specified Packetsの順に移動するときに、キャプチャを保存します。その後、[表示された (Displayed)] パケット範囲のみを保存します。



The image shows the 'Export Specified Packets' dialog box in Wireshark. The 'File name' field is set to 'FTD_Data_only'. The 'Save as type' is set to 'Wireshark/tcpdump/... - pcap (*.dmp.gz;*.dmp;*.cap.gz;*.cap;*.pcap.gz;*.pcap)'. The 'Compress with gzip' checkbox is unchecked. Under the 'Packet Range' section, the 'Displayed' radio button is selected and highlighted with an orange box. The 'All packets' radio button is also selected. The 'Captured' column shows 23988 packets, and the 'Displayed' column shows 23954 packets. Other options include 'Selected packet', 'Marked packets', 'First to last marked', 'Range', and 'Remove ignored packets'.

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:パケット損失の場所を特定します。

このような場合、同時にキャプチャを取得し、分割統治法を使用して、パケット損失の原因となっているネットワークセグメントを特定する必要があります。ファイアウォールの観点では、主なシナリオは次の3つです。

1. パケット損失は、ファイアウォール自体が原因です。
2. パケット損失がファイアウォールデバイスへのダウンストリーム (サーバーからクライアントへの方向) で発生しています。
3. パケット損失がファイアウォールデバイスへのアップストリーム (クライアントからサーバーへの方向) で発生しています。

ファイアウォールによるパケット損失 : パケット損失の原因がファイアウォールであるかどうかを特定するには、入力キャプチャと出力キャプチャを比較する必要があります。2つの異なるキャプチャは、さまざまな方法で比較できます。このセクションでは、このタスクを実行する1つの方法を示します。

パケット損失を特定するために2つのキャプチャを比較する手順

ステップ1 : 2つのキャプチャに同じ時間帯のパケットが含まれていることを確認します。言い換えると、これは、一方のキャプチャに、他方のキャプチャの前後でキャプチャされたパケットが存在していないということです。これは、いくつかの方法で実行できます。

- 最初と最後のパケットについて、そのIP識別子 (ID) の値を確認します。
- 最初と最後のパケットについて、そのタイムスタンプの値を確認します。

この例では、各キャプチャの最初のパケットが持つIP IDの値が同じであることを確認できます

。

No.	Time	Source	Destination	Protocol	Length	Identification	Info
1	2019-10-16 16:13:44.169394	192.168.2.220	192.168.1.220	TCP	74	0xb034 (2612)	54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
2	2019-10-16 16:13:45.195958	192.168.2.220	192.168.1.220	TCP	74	0xb035 (2613)	[TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288526 TSecr=0 WS=128
3	2019-10-16 16:13:47.177542	192.168.2.220	192.168.1.220	TCP	74	0xc151f (5407)	2388 → 54494 [SYN, ACK] Seq=2669989678 Ack=1884231612 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
4	2019-10-16 16:13:47.178030	192.168.2.220	192.168.1.220	TCP	66	0xb036 (2614)	
5	2019-10-16 16:13:47.179647	192.168.2.220	192.168.1.220	TCP	1314	0xc1521 (5409)	
6	2019-10-16 16:13:47.179998	192.168.2.220	192.168.1.220	TCP	66	0xb037 (2615)	
7	2019-10-16 16:13:47.180456	192.168.2.220	192.168.2.220	TCP	1314	0xc1522 (5411)	
8	2019-10-16 16:13:47.1809517	192.168.2.220	192.168.2.220	TCP	1314	0xc1524 (5412)	
9	2019-10-16 16:13:47.180715	192.168.2.220	192.168.1.220	TCP	78	0xb038 (2616)	
10	2019-10-16 16:13:47.180792	192.168.2.220	192.168.1.220	TCP	78	0xb039 (2617)	
11	2019-10-16 16:13:47.489888	192.168.2.220	192.168.2.220	TCP	1314	0xc1525 (5413)	
12	2019-10-16 16:13:47.490376	192.168.2.220	192.168.1.220	TCP	66	0xb03a (2618)	
13	2019-10-16 16:13:47.490865	192.168.2.220	192.168.2.220	TCP	1314	0xc1526 (5414)	
14	2019-10-16 16:13:47.490910	192.168.2.220	192.168.2.220	TCP	1314	0xc1529 (5415)	
15	2019-10-16 16:13:47.490987	192.168.2.220	192.168.2.220	TCP	1314	0xc1529 (5417)	
16	2019-10-16 16:13:47.491231	192.168.2.220	192.168.1.220	TCP	66	0xb03b (2619)	
17	2019-10-16 16:13:47.491261	192.168.2.220	192.168.1.220	TCP	78	0xb03c (2620)	
18	2019-10-16 16:13:47.491261	192.168.2.220	192.168.2.220	TCP	1314	0xc152a (5418)	
19	2019-10-16 16:13:47.492024	192.168.2.220	192.168.2.220	TCP	78	0xb03d (2621)	
20	2019-10-16 16:13:48.410150	192.168.2.220	192.168.2.220	TCP	1314	0xc152e (5422)	
21	2019-10-16 16:13:48.411050	192.168.2.220	192.168.1.220	TCP	66	0xb03e (2622)	
22	2019-10-16 16:13:48.411569	192.168.2.220	192.168.2.220	TCP	1314	0xc152f (5423)	
23	2019-10-16 16:13:48.411630	192.168.2.220	192.168.2.220	TCP	1314	0xc1530 (5424)	
24	2019-10-16 16:13:48.411654	192.168.2.220	192.168.2.220	TCP	1314	0xc1532 (5425)	
25	2019-10-16 16:13:48.411660	192.168.2.220	192.168.2.220	TCP	1314	0xc1533 (5427)	
26	2019-10-16 16:13:48.411859	192.168.2.220	192.168.1.220	TCP	66	0xb03f (2623)	
27	2019-10-16 16:13:48.412088	192.168.2.220	192.168.1.220	TCP	66	0xb040 (2624)	
28	2019-10-16 16:13:48.410074	192.168.2.220	192.168.2.220	TCP	1314	0xc152e (5422)	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224
29	2019-10-16 16:13:48.411074	192.168.2.220	192.168.1.220	TCP	66	0xb03e (2622)	54494 → 2388 [ACK] Seq=2157
30	2019-10-16 16:13:48.411538	192.168.2.220	192.168.2.220	TCP	1314	0xc152f (5423)	2388 → 54494 [ACK] Seq=2224
31	2019-10-16 16:13:48.411599	192.168.2.220	192.168.2.220	TCP	1314	0xc1530 (5424)	2388 → 54494 [ACK] Seq=2224

それらが同じでない場合は、次の手順を実行します。

1. 各キャプチャの最初の packets でタイムスタンプを比較します。
2. 一番最後のタイムスタンプを持つキャプチャでフィルタを用意し、タイムスタンプフィルタを [=] から >=] (最初の packets) および <=] (最後の packets) に変更します。たとえば、次のようになります。

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-16 16:13:43.244692	192.168.2.220	192.168.1.220	TCP	74	38400 → 21 [S
2	2019-10-16 16:13:43.245638	192.168.1.220	192.168.2.220	TCP	74	21 → 38400 [S
3	2019-10-16 16:13:43.245867	192.168.2.220	192.168.1.220	TCP	66	38400 → 21 [A

▼ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 Encapsulation type: Ethernet (1)
 Arrival Time: Oct 16, 2019 16:13:43.245638000
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1571235223.245638000 seconds
 [Time delta from previous captured frame: 0.000000000 seconds]
 [Time delta from previous displayed frame: 0.000000000 seconds]
 [Time since reference or first frame: 0.000900000 seconds]
 Frame Number: 2
 Frame Length: 74 bytes (592 bits)
 Capture Length: 74 bytes (592 bits)

- Expand Subtrees
- Collapse Subtrees
- Expand All
- Collapse All
- Apply as Column
- Apply as Filter
- Prepare a Filter

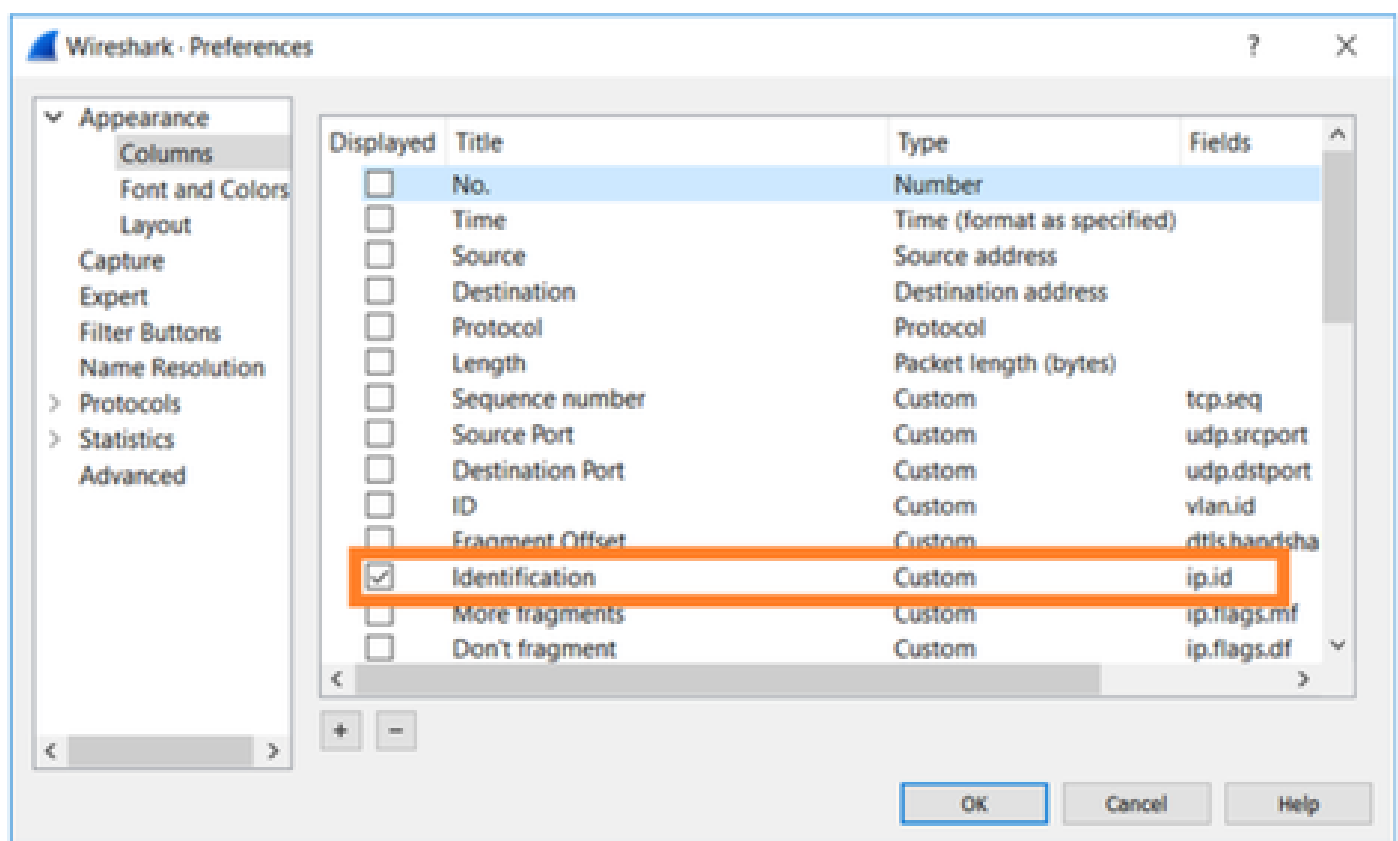
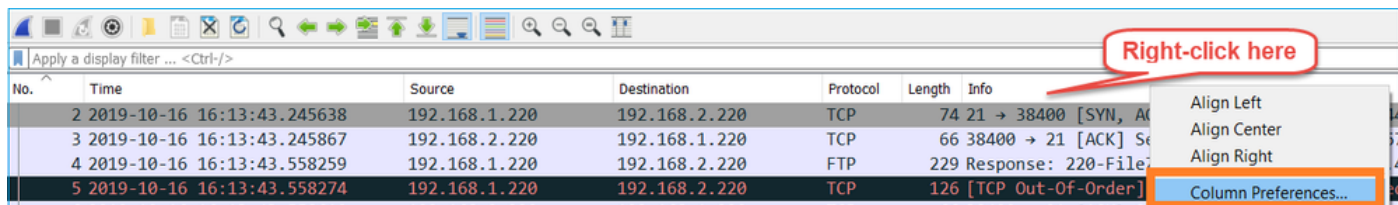
(frame.time >= "Oct 16, 2019 16:13:43.244692000") &&(frame.time <= "Oct 16, 2019 16:20:21.785130000")

3. 指定した packets を新しいキャプチャにエクスポートします。[ファイル (File)] > [指定した packets のエクスポート (Export Specified Packets)] を選択し、[表示された (Displayed)] packets を保存してください。この時点で、両方のキャプチャに、同じ時間枠を対象とする packets が含まれている必要があります。これで、2つのキャプチャの比較を開始できます。

ステップ 2: 2つのキャプチャ間の比較に使用する packets フィールドを指定します。使用できるフィールドの例:

- IP ID
- RTP シーケンス番号
- ICMP シーケンス番号

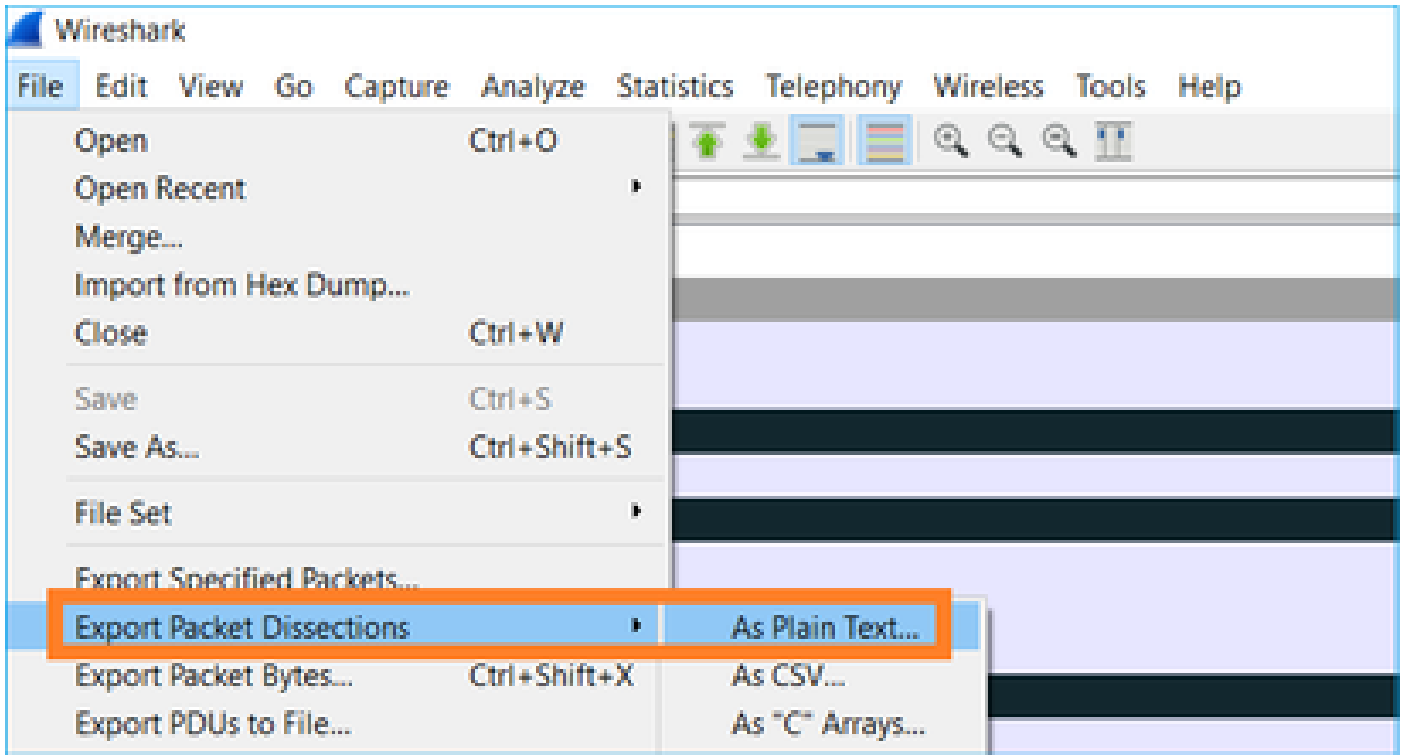
ステップ1で指定した各パケットのフィールドを含む各キャプチャのテキストバージョンを作成します。これを行うには、対象の列のみを残します。たとえば、IP IDに基づいてパケットを比較する場合は、図に示すようにキャプチャを変更します。



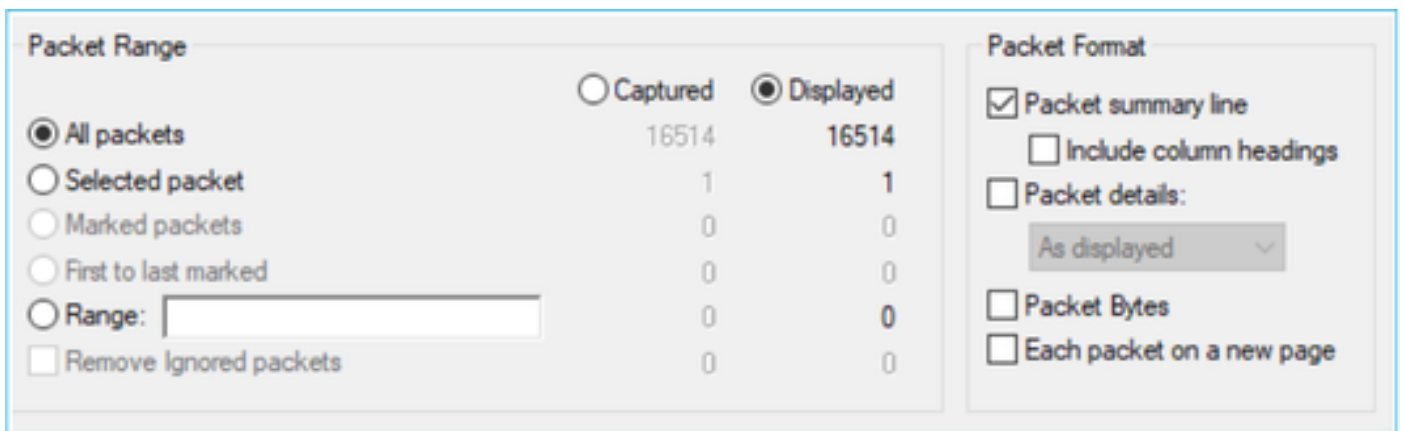
結果は、次のとおりです。

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

ステップ 3 : 図に示すように、キャプチャのテキストバージョンを作成します(File > Export Packet Dissections > As Plain Text...)



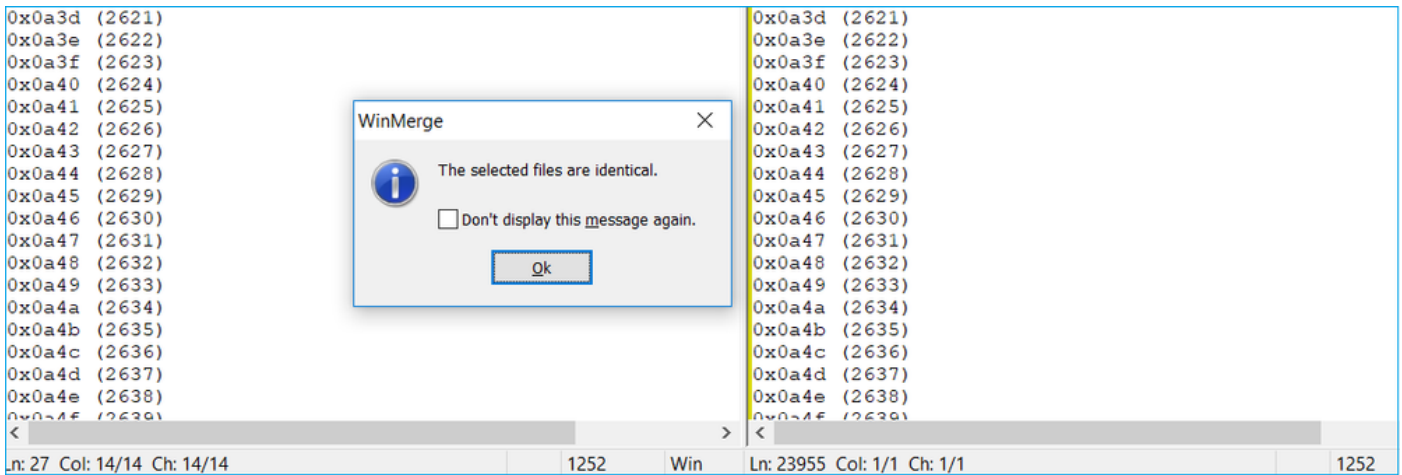
表示されたフィールドの値のみをエクスポートするには、図のように、[カラムヘッダーを含める (Include column headers)] オプションと [パケットの詳細 (Packet details)] オプションをオフにします。



ステップ 4 : ファイル内のパケットを並べ替えます。これを行うには、Linux の sort コマンドを使用します。

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


ステップ 5 : テキスト比較ツール (WinMerge など) または Linux の diff コマンドを使用して、2 つのキャプチャの違いを見つけます。



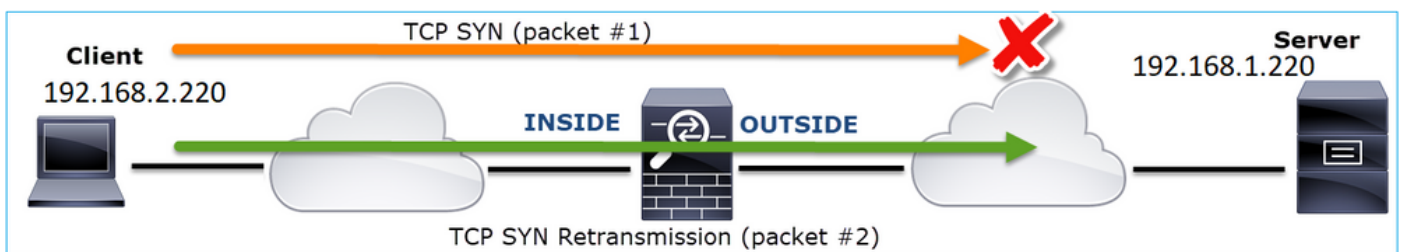
この場合、FTP データトラフィックの CAPI および CAPO キャプチャは同一です。これは、パケット損失の原因がファイアウォールではないことの証明となります。

アップストリーム/ダウンストリームのパケット損失を識別します。

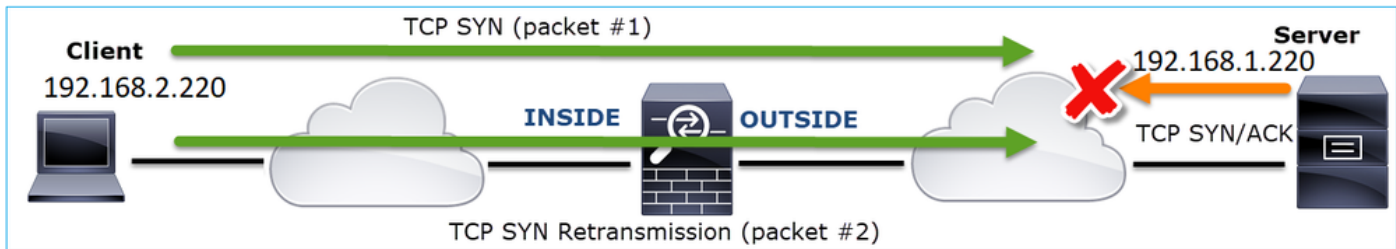
No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-16 16:13:44.169516	192.168.2.220	192.168.1.220	TCP	74	54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=1
2	2019-10-16 16:13:45.196050	192.168.2.220	192.168.1.220	TCP	74	[TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577291510 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=3577291510 TSecr=3577291510
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=3577291820
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=3577291510
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=3577291510
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=3577291820
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

キーポイント :

1. このパケットは TCP 再送信です。具体的には、パッシブモードの FTP データのためにクライアントからサーバーに送信される TCP SYN パケットです。クライアントがパケットを再送信しており、最初の SYN (パケット番号 1) が確認できるため、パケットはファイアウォールへのアップストリームで失われています。



この場合、SYNパケットはサーバに到達したものの、SYN/ACKパケットが戻る途中で失われた可能性があります。



2. サーバーからのパケットが存在し、Wireshark は前のセグメントが確認/キャプチャされていないことを識別しています。キャプチャされていないパケットは、サーバーからクライアントに送信されていますが、ファイアウォールキャプチャでは確認されていません。つまり、パケットはサーバーとファイアウォールの間で失われています。



これは、FTP サーバーとファイアウォールの間にパケット損失があることを示しています。

アクション2:追加のキャプチャを取得します。

エンドポイントでのキャプチャとともに、追加のキャプチャを取得します。分割統治法を適用して、パケット損失の原因となっている問題のあるセグメントの絞り込みを試みます。

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)

```

キーポイント：

1. 受信者（この場合は FTP クライアント）は、着信 TCP シーケンス番号を追跡しています。パケットが失われた（予期されたシーケンス番号がスキップされた）ことを検出すると、ACK=「予期されていたもののスキップされたシーケンス番号」の ACK パケットを生成し

ます。この例では、Ack=2224386800 です。

2. Dup ACKにより、TCP高速再送信 (Duplicate ACK受信後20ミリ秒以内の再送信) がトリガーされます。

重複 ACK は、次のことを意味します。

- 複数回の重複 ACK があり、実際の再送信がない場合は、到着順序が不正なパケットが存在する可能性が高いことを示しています。
- 実際の再送信に続く重複 ACK は、ある程度のパケット損失があることを示しています。

アクション3:中継パケットのファイアウォール処理時間を計算します。

2つの異なるインターフェイスに同じキャプチャを適用します。

```
<#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
```

キャプチャをエクスポートし、入力パケットと出力パケットの時間差を確認します。

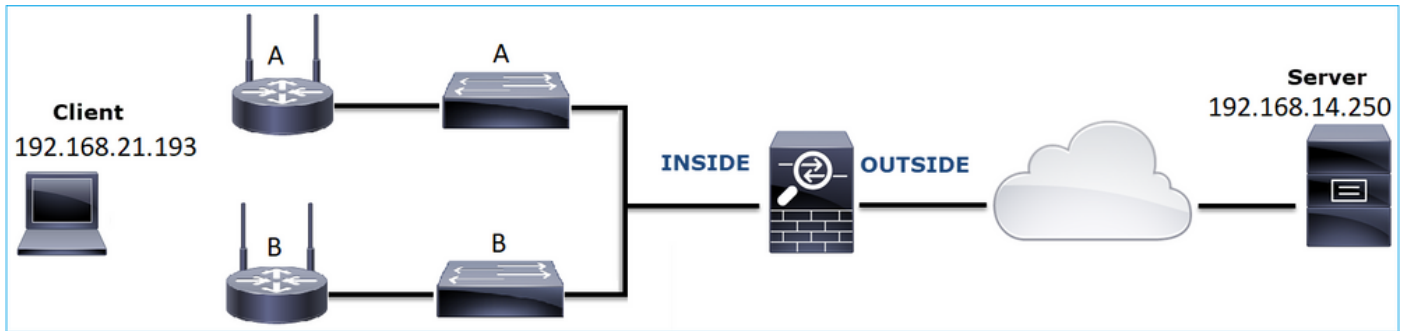
Case 7.TCP接続の問題 (パケット破損)

事象の説明:

ワイヤレスクライアント (192.168.21.193) は宛先サーバー (192.168.14.250 : HTTP) への接続を試み、次の2つの異なるシナリオが存在します。

- クライアントがアクセスポイント (AP) 「A」に接続すると、HTTP 接続が機能しません。
- クライアントがアクセスポイント (AP) 「B」に接続すると、HTTP 接続が機能します。

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:192.168.21.193

宛先IP:192.168.14.250

プロトコル：TCP 80

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

```
<#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
```

キャプチャ - 機能シナリオ：

ベースラインとして、既知の正常なシナリオからキャプチャを取得すると常に非常に便利です。

次の図は、NGFW の INSIDE インターフェイスで取得されたキャプチャを示しています。

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

次の図は、NGFW の OUTSIDE インターフェイスで取得されたキャプチャを示しています。

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

キーポイント：

1. 2つのキャプチャはほぼ同じです (ISN のランダム化を考慮) 。
2. パケット損失の兆候は存在しません。
3. 順序不正 (OOO) パケットは存在しません。
4. 3つのHTTP GET 要求が存在します。最初のもは 404 「Not Found」、2つ目のものは 200 「OK」、3つ目のものは 304 「Not Modified」リダイレクトメッセージを受け取っています。

キャプチャ：既知の障害があるシナリオ：

入力キャプチャ (CAPI) の内容は、次のとおりです。

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	192.168.21.193	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

キーポイント：

1. TCP 3ウェイハンドシェイクが存在します。
2. TCP 再送信があり、パケット損失の兆候が存在します。
3. Wireshark によって「Malformed」(不正)として識別されたパケット (TCP ACK) が存在します。

この図は、出力キャプチャ (CAPO) の内容を示しています。

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	192.168.21.193	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=268013987 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=2731219423 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

キーポイント：

2つのキャプチャはほぼ同じです (ISN のランダム化を考慮) 。

1. TCP 3ウェイハンドシェイクが存在します。
2. TCP 再送信があり、パケット損失の兆候が存在します。
3. Wireshark によって「Malformed」(不正)として識別されたパケット (TCP ACK) が存在します。

不正なパケットを確認します。

```
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] 1

> Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> 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
> Transmission Control Protocol, Src Port: 3072, Dst Port: 80, Seq: 4231766829, Ack: 867575960, Len: 2 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) 3
  [Malformed Packet: Tunnel Socket] 1
  [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 00 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 4  (.P.....-..)
```

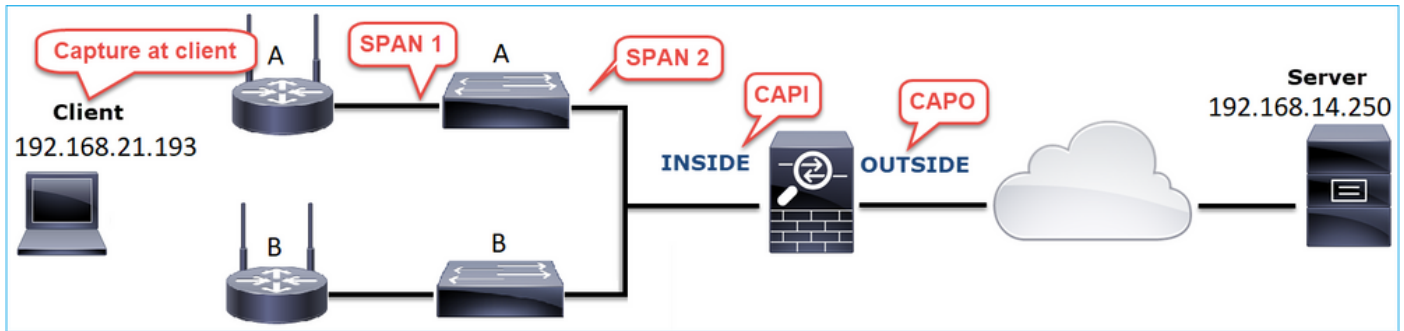
キーポイント：

1. パケットは Wireshark によって「Malformed」(不正)と識別されています。
2. 長さは 2 バイトです。
3. 2 バイトの TCP ペイロードが存在します。
4. ペイロードは 4 つの追加のゼロ (00 00) です。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:追加のキャプチャを取得します。エンドポイントでキャプチャを含め、可能であれば、パケット破損の原因を切り分けるために分割統治法を適用してみてください。次に例を示します。

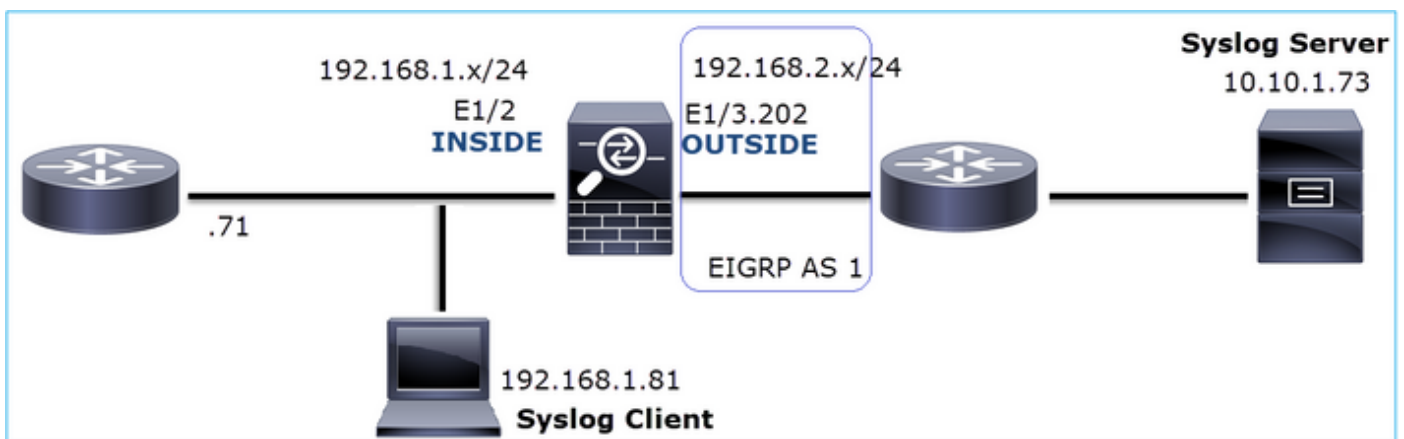


この場合、スイッチ「A」のインターフェイスドライバによって2つの追加バイトが付加されており、解決策は破損の原因となっているスイッチを交換することでした。

Case 8.UDP接続の問題 (欠落パケット)

問題の説明：宛先syslogサーバにsyslog(UDP 514)メッセージが表示されない。

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:192.168.1.81

宛先IP:10.10.1.73

プロトコル：UDP 514

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

```
<#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 キャプチャにはパケットが表示されていません。

```
<#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
```

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:FTD接続テーブルをチェックします。

特定の接続を確認するには、次の構文を使用します。

```
<#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
```

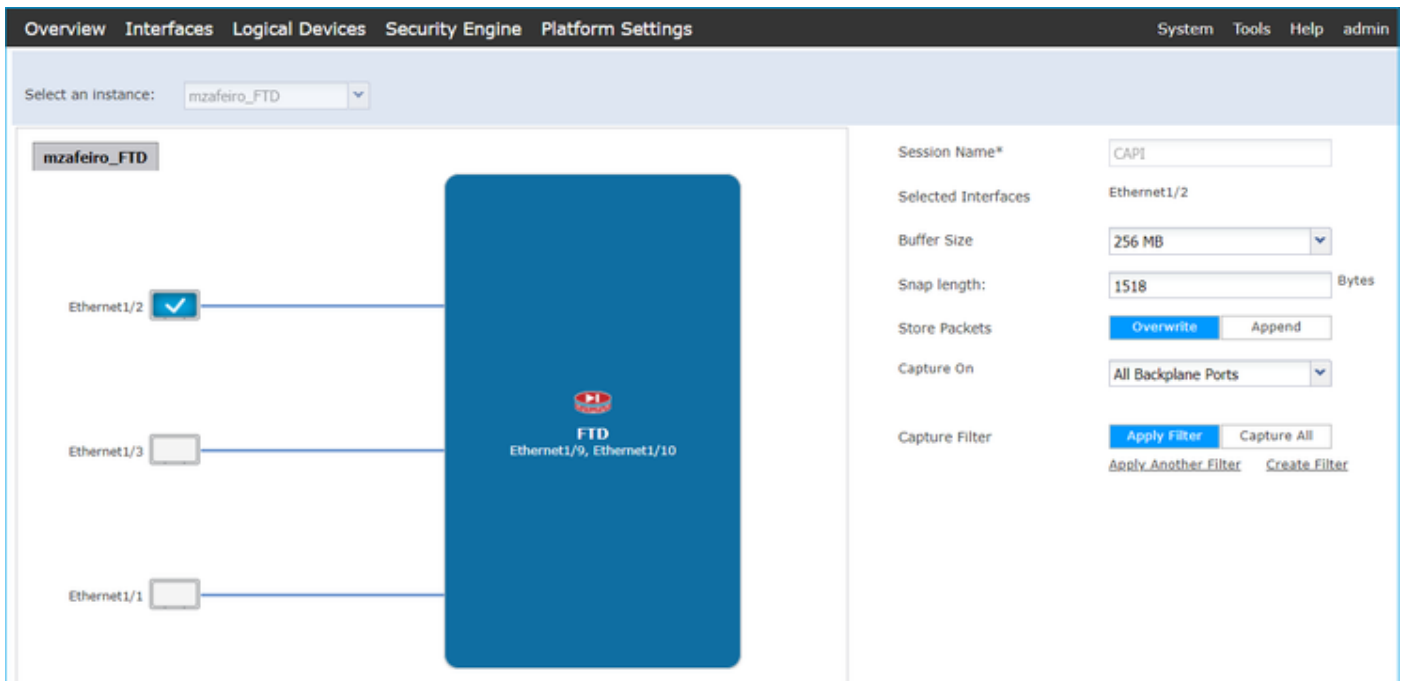
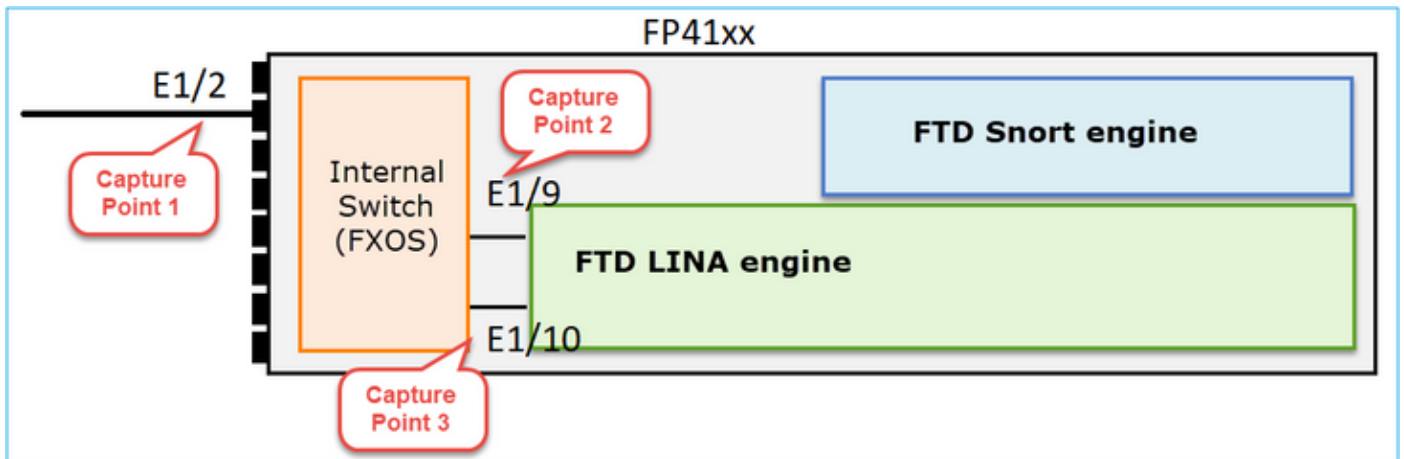
```
N1
```

キーポイント：

1. 入インターフェイスと出インターフェイスが同じです (Uターンしています)。
2. バイト数が非常に大きな値 (約 5 GB) になっています。
3. フラグ「o」は、フローオフロード (ハードウェア アクセラレーション フロー) を示します。これが、FTD キャプチャにパケットが表示されない理由です。フローオフロードは、41xx プラットフォームと 93xx プラットフォームでのみサポートされています。この場合、デバイスは 41xx です。

アクション2:シャーシレベルのキャプチャを取得します。

次の図のように、Firepower のシャーシマネージャに接続し、入インターフェイス (この場合は E1/2) およびバックプレーン インターフェイス (E1/9 と E1/10) でのキャプチャを有効にします。



数秒後に、次のようになります。

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

🔍 ヒント:Wiresharkでは、VNタグ付きパケットを除外して、物理インターフェイスレベルでのパケットの重複を排除します

変更前 :

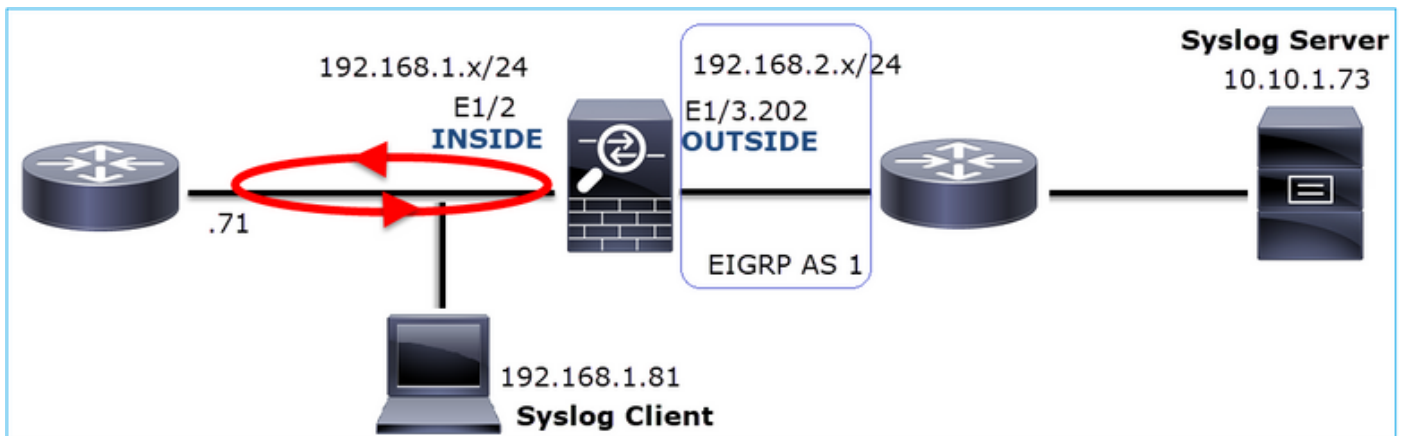
No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000..	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
2	0.0000..	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
3	0.0532..	Vmware_85:4f:ca	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

変更後 :

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	254	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

キーポイント：

1. 表示フィルタが適用され、パケットの重複が削除されて、Syslog のみが表示されています。
2. パケット間の時間差はマイクロ秒レベルです。これは、パケットレートが非常に高いことを示しています。
3. 存続可能時間 (TTL) の値が継続的に減少しています。これは、パケットのループを示しています。



アクション3:パケットトレーサを使用します。

パケットがファイアウォールの LINA エンジンを通過しないため、ライブトレース (トレース付きのキャプチャ) は実行できませんが、パケットトレーサを使用してエミュレートされたパケットをトレースできます。

```
<#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
```

アクション4:FTDルーティングを確認します。

ファイアウォール ルーティング テーブルを調べて、ルーティングに問題がないか確認します。

```
<#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
```

キー ポイント :

1. ルートは正しい出カインターフェイスに向かっています。
2. ルートは数分前 (0:02:37) に学習されています。

アクション5:接続の稼働時間を確認します。

接続の継続時間を調べて、この接続がいつ確立されたのかを確認します。

```
<#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
```

重要なポイント :

1. 接続は約 4 分前 (ルーティングテーブルに EIGRP ルートがインストールされる前) に確立されています。

アクション6:確立された接続をクリアします。

この場合、パケットは確立された接続に一致し、誤った出カインターフェイスにルーティングされます。これによりループが発生します。これは、ファイアウォールの次の動作順序が原因です。

1. 確立された接続のルックアップ (これは、グローバル ルーティング テーブルのルックアップよりも優先されます)
2. ネットワークアドレス変換 (NAT) のルックアップ (UN-NAT (宛先 NAT) フェーズは、PBR およびルートのルックアップよりも優先されます)
3. ポリシーベース ルーティング (PBR)
4. グローバル ルーティング テーブルのルックアップ

接続は決してタイムアウトしない (UDP 接続がアイドルタイムアウトする 2 分間の間に Syslog クライアントがパケットを継続的に送信します) ため、接続を手動でクリアする必要があります。

```
<#root>
```

```
firepower#
```

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

```
1 connection(s) deleted.
```

新しい接続が確立されることを確認します。

```
<#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
```

アクション7:フローティングコネクトタイムアウトを設定します。

これは、特に UDP フローについて、問題に対処し、最適でないルーティングを回避するために適切なソリューションです。[デバイス (Devices)] > [プラットフォーム設定 (Platform Settings)] > [タイムアウト (Timeouts)] に移動し、値を設定します。

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)

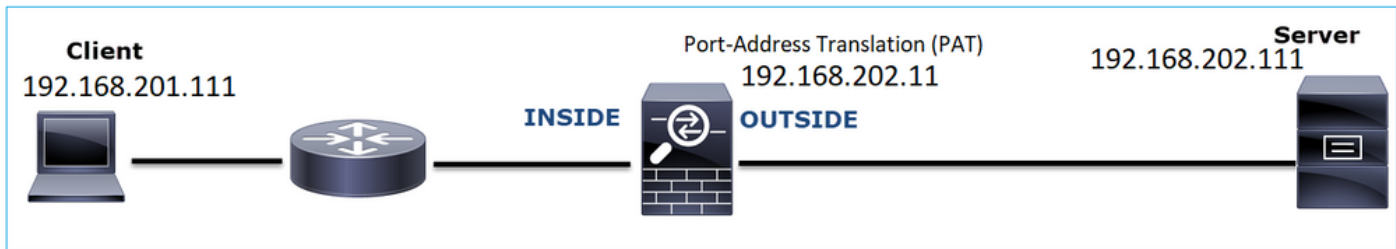
フローティング接続タイムアウトの詳細については、コマンドリファレンスを参照してください。

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

Case 9.HTTPS接続の問題 (シナリオ1)

問題の説明 : クライアント192.168.201.105とサーバ192.168.202.101間のHTTPS通信が確立できない

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:192.168.201.111

宛先IP:192.168.202.111

プロトコル：TCP 443(HTTPS)

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

OUTSIDE キャプチャで使用される IP は、ポートアドレス変換の設定により異なります。

```
<#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
```

次の図は、NGFW の INSIDE インターフェイスで取得されたキャプチャを示しています。

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 10:39:35.188909	192.168.202.111	192.168.201.111	TCP	78	0x0000 (0)	443 → 6666 [SYN, ACK] Seq=4086514531 Ack=2034865632 Win=28960 Len=0 MSS=1380 SACK_PERM=1 TSval=3119615816 TSecr=192658158
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

キーポイント：

1. TCP 3 ウェイハンドシェイクが存在します。
2. SSL ネゴシエーションが開始されています。クライアントが Client Hello メッセージを送信しています。
3. クライアントに送信された TCP ACK が存在します。
4. クライアントに送信された TCP RST が存在します。

次の図は、NGFW の OUTSIDE インターフェイスで取得されたキャプチャを示しています。

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	0x0000 (0)	443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=192660198
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 10:39:37.274830	192.168.202.11	192.168.202.111	TCP	257	0x0095 (47305)	[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	0x08af (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	0x0621 (54817)	15880 → 443 [RST] Seq=2486930895 Win=0 Len=0 TSval=192688266 TSecr=0
41	2018-02-01 10:40:06.790700	192.168.202.111	192.168.202.11	TCP	78	0x0000 (0)	[TCP Retransmission] 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=192660198

キーポイント：

1. TCP 3 ウェイハンドシェイクが存在します。
2. SSL ネゴシエーションが開始されています。クライアントが Client Hello メッセージを送信しています。
3. ファイアウォールからサーバーに送信された TCP 再送信が存在します。
4. サーバーに送信された TCP RST が存在します。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

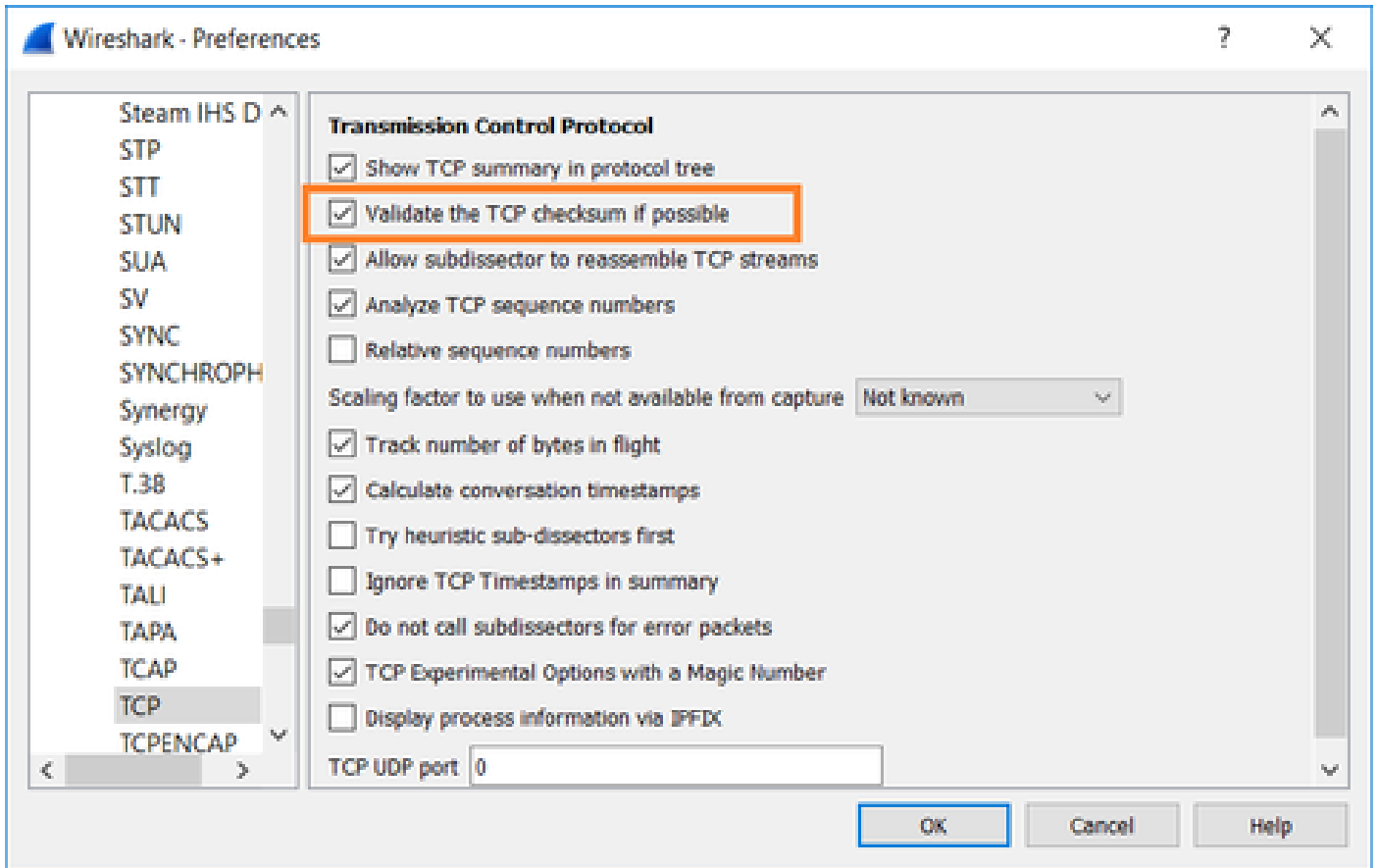
アクション1:追加のキャプチャを取得します。

サーバーで取得したキャプチャから、サーバーが破損した TCP チェックサムをとまなう TLS Client Hello を受信しており、それらをサイレントにドロップしたことがわかります (クライアントへの TCP RST またはその他の応答パケットは存在しません)。

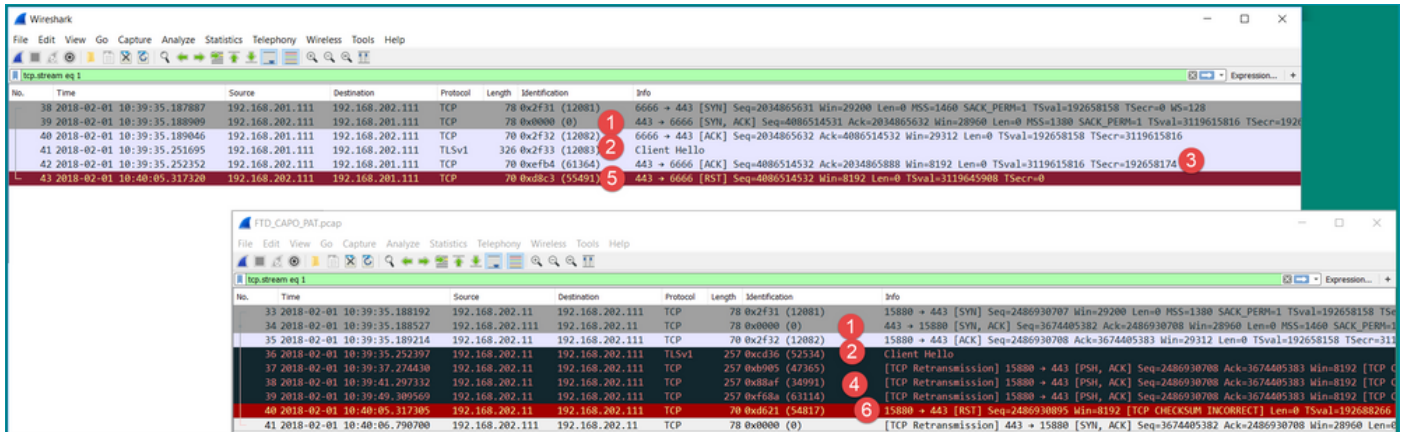
```
21:26:27.133677 IP (tos 0x0, ttl 64, id 52534, offset 0, flags [DF], proto TCP (6), length 239)
 192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x0c65 (incorrect -> 0x3063), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192658174 ecr 3119615816], length 187
21:26:29.155652 IP (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 0x3dd (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
```

すべてを組み合わせると、次のように結論付けられます。

この場合、理解するために、Wiresharkで「Validate the TCP checksum if possible」オプションを有効にする必要があります。次の図のように、[編集 (Edit)] > [設定 (Preferences)] > [プロトコル (Protocols)] > [TCP] に移動します。



この場合、全体像を把握するためにキャプチャを並べて配置すると便利です。



キーポイント：

1. TCP 3ウェイハンドシェイクが存在します。IP ID は同じです。これは、フローがファイアウォールによってプロキシされなかったことを意味します。
2. TLS Client Helloは、IP ID 12083のクライアントから送信されます。パケットはファイアウォールによってプロキシされ (この場合、ファイアウォールはTLS復号化ポリシーで設定されています)、IP IDは52534に変更されます。また、パケットのTCPチェックサムが破損します(後で修正されたソフトウェア不具合が原因)。
3. ファイアウォールはTCPプロキシモードで、クライアント (サーバをスプーフィングする) にACKを送信します。


```

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 Min=29200 Len=0 MSS=1380 S
34 2018-02-01 10:39:35.188527 192.168.202.111 192.168.202.11 TCP 78 0x0000 (0) 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Min=29
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 Min=29312 L
36 2018-02-01 10:39:35.252397 192.168.202.11 192.168.202.111 TLSv1 257 0xcd36 (52534) Client Hello

```

```

> Internet Protocol Version 4, Src: 192.168.202.11, Dst: 192.168.202.111
  Transmission Control Protocol, Src Port: 15880, Dst Port: 443, Seq: 2486930708, Ack: 3674405383, Len: 187
    Source Port: 15880
    Destination Port: 443
    [Stream index: 1]
    [TCP Segment Len: 187]
    Sequence number: 2486930708
    [Next sequence number: 2486930895]
    Acknowledgment number: 3674405383
    1000 ... = Header Length: 32 bytes (8)
    > Flags: 0x018 (PSH, ACK)
    Window size value: 64
    [Calculated window size: 8192]
    [Window size scaling factor: 128]
    > Checksum: 0x0c65 incorrect, should be 0x3063(maybe caused by "TCP checksum offload?")
    [Checksum Status: Bad]
    [Calculated Checksum: 0x3063]
    Urgent pointer: 0
    > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    > [SEQ/ACK analysis]
    > [Timestamps]
    TCP payload (187 bytes)
  Secure Sockets Layer

```

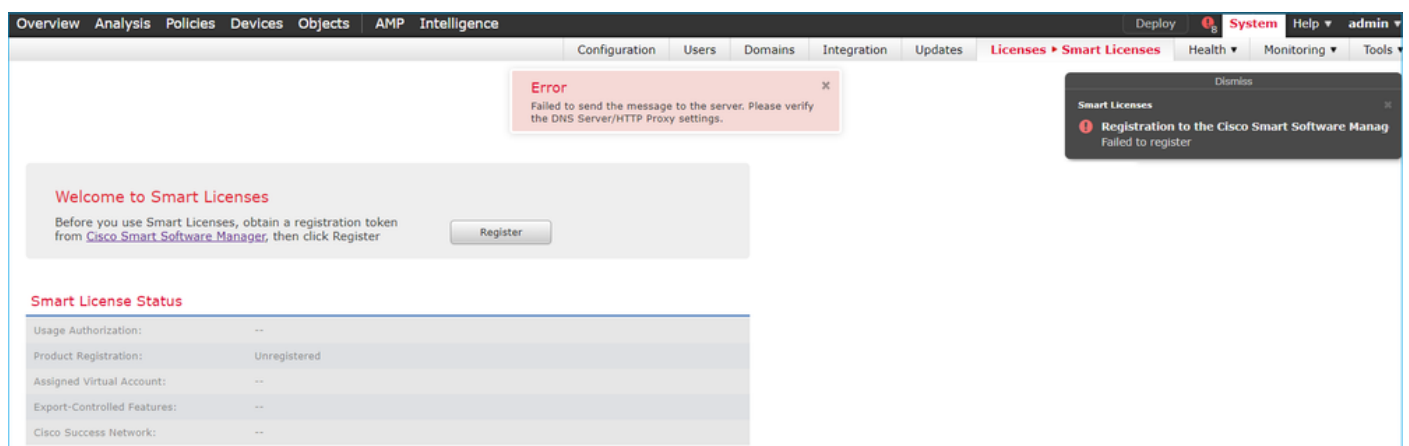
4. ファイアウォールはサーバーから TCP ACK パケットを受信しておらず、TLS Client Hello メッセージを再送信しています。これも、ファイアウォールがアクティブ化した TCP プロキシモードが原因です。
5. 約 30 秒後、ファイアウォールは中断し、TCP RST をクライアントに送信しています。
6. ファイアウォールがサーバーに向けて TCP RST を送信しています。

次のドキュメントを参照してください。

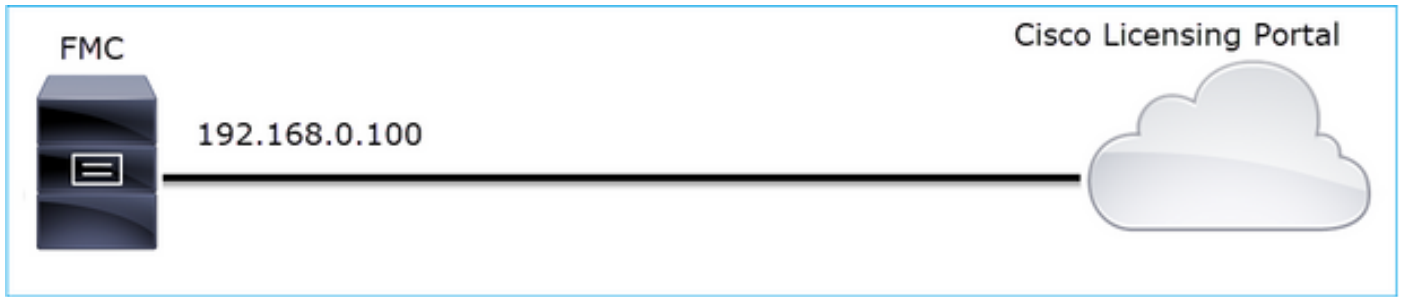
[Firepower の TLS/SSL ハンドシェイク処理](#)

Case 10.HTTPS接続の問題 (シナリオ2)

問題の説明 : FMC スマートライセンスの登録に失敗します。



次の図は、このトポロジを示しています。



影響を受けるフロー :

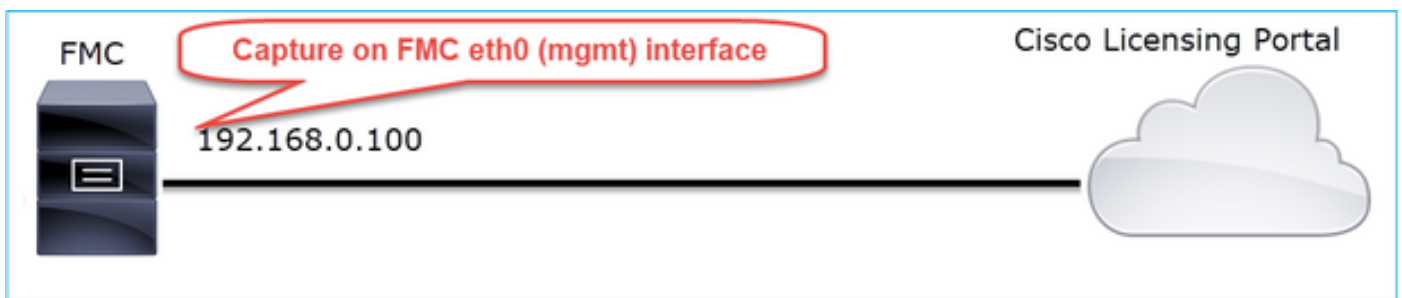
送信元IP:192.168.0.100

宛先 : tools.cisco.com

プロトコル : TCP 443(HTTPS)

キャプチャ分析

FMC 管理インターフェイスでのキャプチャを有効にします。



登録を再試行します。エラーメッセージが表示されたら、Ctrl + C キーを押してキャプチャを停止します。

```
<#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
```

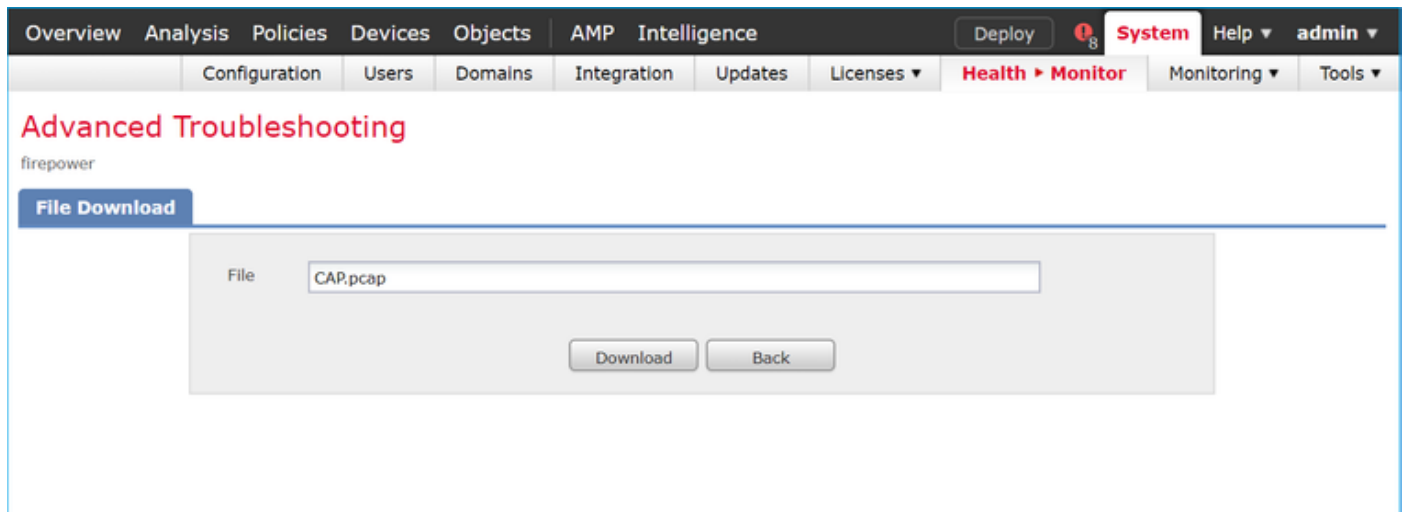
```
<- CTRL-C
```

```
264 packets received by filter
```

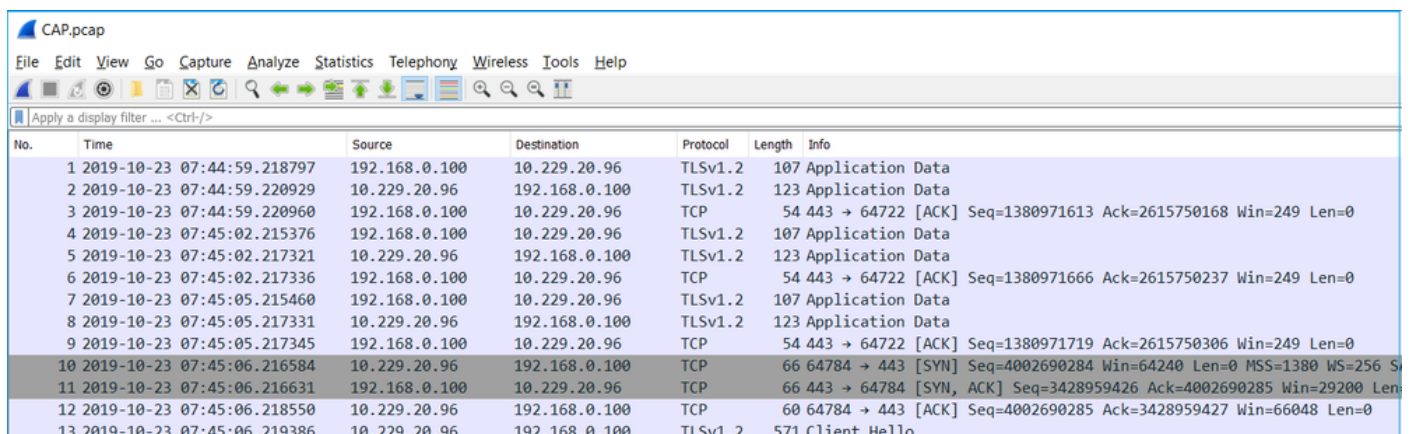
```
0 packets dropped by kernel
```


```
root@firepower:/Volume/home/admin#
```

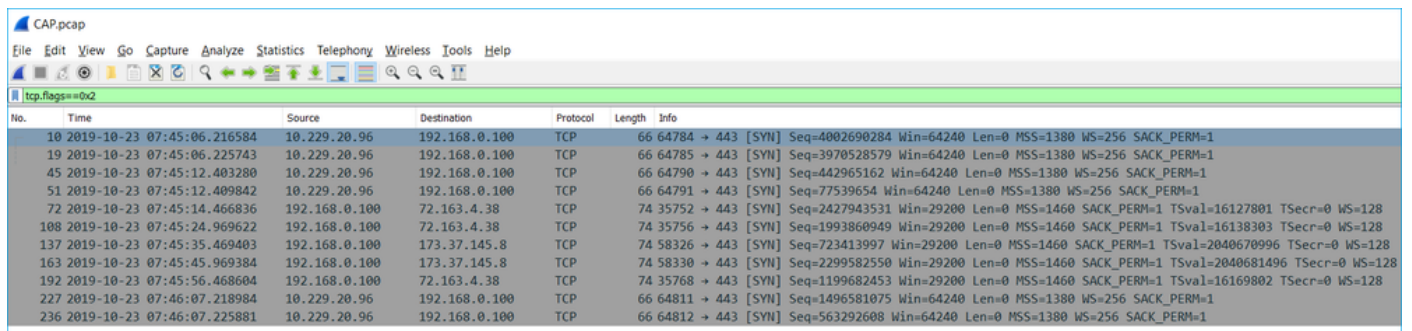
次の図のように、FMC からキャプチャを収集します ([システム (System)] > [ヘルス (Health)] > [モニター (Monitor)] に移動し、デバイスを選択して、[高度なトラブルシューティング (Advanced Troubleshooting)] を選択します)。



次の図は、Wireshark での FMC キャプチャを示しています。



 ヒント：キャプチャされたすべての新しいTCPセッションを確認するには、Wiresharkで tcp.flags==0x2 表示フィルタを使用します。これにより、キャプチャされたすべての TCP SYN パケットがフィルタ処理されます。



 ヒント:SSL Client HelloのServer Nameフィールドを列として適用します。

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

ヒント：この表示フィルタを適用すると、Client Helloメッセージssl.handshake.type == 1だけが表示されます。

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

注：本書の執筆時点では、スマートライセンスポータル(tools.cisco.com)は72.163.4.38、173.37.145.8のIPを使用しています。

次の図のように、いずれかの TCP フローを追跡します ([追跡 (Follow)] > [TCPストリーム (TCP Stream)])。

75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.co
111	2019-10-23 07:45:25.136089	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.co
140	2019-10-23 07:45:35.637252	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.co
166	2019-10-23 07:45:46.136858	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.co
195	2019-10-23 07:45:56.635438	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.co
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	

rame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits)
 thernet II, Src: Vmware_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
 nternet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38
 ransmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 517
 eecure Sockets Layer
 TLSv1.2 Record Layer: Handshake Protocol: Client Hello
 Content Type: Handshake (22)
 Version: TLS 1.0 (0x0301)
 Length: 512

- 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 Conversion
- SCTP
- Follow
 - TCP Stream
 - UDP Stream
 - SSL Stream
 - HTTP Stream
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

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 TSecr=0 WS=128
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	1304		443 → 35752 [PSH, ACK] Seq=2770078981 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 07:45:14.966885	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 07:45:14.967201	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [RST, ACK] Seq=2427944056 Ack=2770080421 Win=0 Len=0
87	2019-10-23 07:45:14.967282	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)
 > 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: 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: 234490a107438c73b58564653271c7c09fbb7ac16897184...
 Session ID Length: 0
 Cipher Suites Length: 100
 Cipher Suites (50 suites)

キーポイント：

1. TCP 3ウェイハンドシェイクが存在します。
2. クライアント (FMC) が SSL Client Hello メッセージをスマートライセンスポータルに送信しています。
3. SSLセッションIDは0です。これは、セッションが再開されていないことを意味します。
4. 宛先サーバーが Server Hello、Certificate、および Server Hello Done メッセージで応答しています。
5. クライアントは、「不明なCA」に関するSSL致命的アラートを送信します。
6. クライアントが TCP RST を送信してセッションを終了しています。
7. TCP セッションの時間は全体 (確立から終了まで) で約 0.5 秒でした。

サーバーの証明書 (Certificate) を選択し、[発行者 (issuer)] フィールドを展開して、commonName を表示します。この場合、この共通名は、中間者 (MITM) を実行しているデバイスを示しています。

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	TLV1.2	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	TLV1.2	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	TLV1.2	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-sta
                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)
                        > RDNSquence item: 1 item (id-at-organizationName=FTD_O)
                        > RDNSquence item: 1 item (id-at-organizationalUnitName=FTD_OU)
                        > RDNSquence item: 1 item (id-at-commonName=FTD4100_MITM)
                  > validity
                  > subject: rdnSequence (0)
                  > subjectPublicKeyInfo
                > extensions: 6 items
  
```

次の図は、このことを示しています。

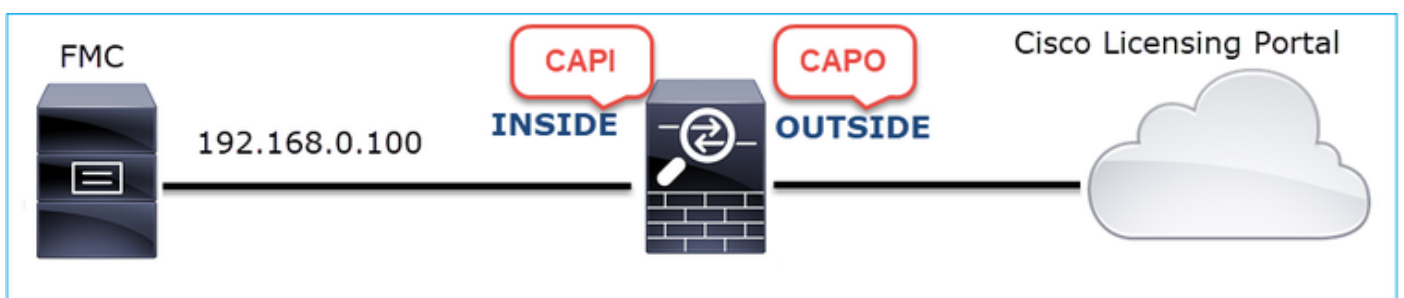


推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:追加のキャプチャを取得します。

中継ファイアウォールデバイスでキャプチャを取得します。



CAPI の内容は、次のとおりです。

```

tcp.stream eq 57
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=1336
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
    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

```

CAPO の内容は、次のとおりです。

```

tcp.stream eq 57
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=
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 Win=8190 Len=0 MSS=1336
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
    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

```

これらのキャプチャから、中継ファイアウォールがサーバー証明書を変更 (MITM) したことが分かります。

アクション2: デバイスログを確認します。

このドキュメントで説明するように、FMC TS バンドルを収集できます。

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

[SourceFire-00.html](#)

この場合、/dir-archives/var-log/process_stdout.log ファイルに次のようなメッセージが表示されます。

```
<#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/
```

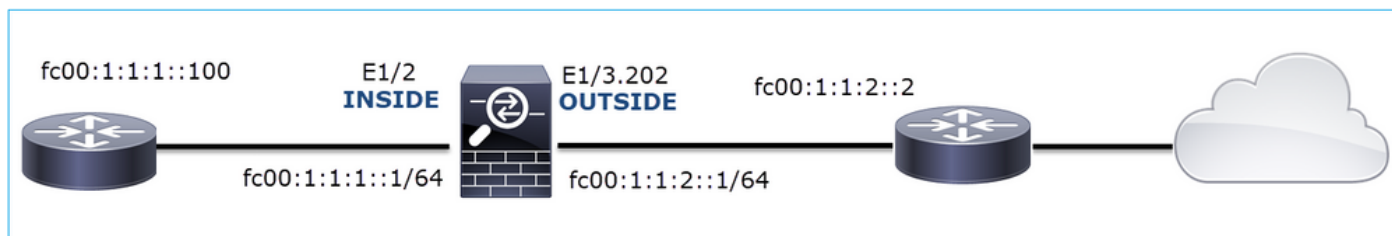
推奨される解決策

特定のフローの MITM を無効にして、FMC がスマートライセンスクラウドに正常に登録できるようにします。

Case 11.IPv6接続の問題

問題の説明：内部ホスト（ファイアウォールのINSIDEインターフェイスの背後にある）は外部ホスト（ファイアウォールのOUTSIDEインターフェイスの背後にあるホスト）と通信できません。

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:fc00:1:1:1::100

宛先IP:fc00:1:1:2::2

プロトコル：任意

キャプチャ分析

FTD LINA エンジンでのキャプチャを有効にします。

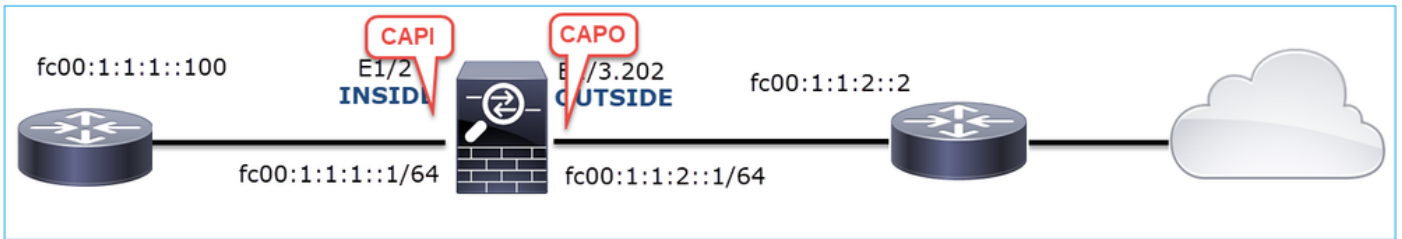
```
<#root>
```

```
firepower#
```

```
capture CAPI int INSIDE match ip any6 any6
```

```
firepower#
```

```
capture CAPO int OUTSIDE match ip any6 any6
```



キャプチャ - 非機能シナリオ :

これらのキャプチャは、IP fc00:1:1:1::100 (内部ルータ) から IP fc00:1:1:2::2 (アップストリームルータ) への ICMP 接続テストと並行して取得されています。

ファイアウォールの INSIDE インターフェイスでのキャプチャには、次の内容が含まれています

。

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:fefc:fcd8	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:fefc:fcd8	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:fefc:fcd8	ICMPv6	86	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8 from 00:be:75:f6:1d:ae
13	2019-10-24 13:02:22.063000	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1dae	ICMPv6	78	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

キーポイント :

1. ルータはIPv6ネイバー送信要求メッセージを送信し、アップストリームデバイスのMACアドレス(IP fc00:1:1:1::1)を要求します。
2. ファイアウォールが IPv6 の Neighbor Advertisement で応答しています。
3. ルータが ICMP エコー要求を送信しています。
4. ファイアウォールはIPv6ネイバー送信要求メッセージを送信し、ダウンストリームデバイスのMACアドレス(fc00:1:1:1::100)を要求します。
5. ルータが IPv6 の Neighbor Advertisement で応答しています。
6. ルータが追加の IPv6 ICMP エコー要求を送信しています。

ファイアウォールの OUTSIDE インターフェイスでのキャプチャには、次の内容が含まれていません。

No.	Time	Source	Destination	Protocol	Port	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)

キーポイント：

1. ファイアウォールは、アップストリームデバイスのMACアドレス(IP fc00:1:1:2::2)を要求するIPv6ネイバー送信要求メッセージを送信します。
2. ルータが IPv6 の Neighbor Advertisement で応答しています。
3. ファイアウォールが IPv6 ICMP エコー要求を送信しています。
4. アップストリームデバイス (ルータfc00:1:1:2::2) は、IPv6アドレスfc00:1:1:1::100のMACアドレスを要求するIPv6ネイバー送信要求メッセージを送信します。
5. ファイアウォールが追加の IPv6 ICMP エコー要求を送信しています。
6. アップストリームルータは、追加のIPv6ネイバー送信要求メッセージを送信し、IPv6アドレスfc00:1:1:1::100のMACアドレスを要求します。

ポイント 4 は非常に興味深い点です。通常、アップストリームルータはファイアウォールの OUTSIDE インターフェイス(fc00:1:1:2::2)のMACを要求しますが、代わりにfc00:1:1:1::100を要求します。これは設定ミスを示しています。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:IPv6ネイバーテーブルをチェックします。

ファイアウォールの IPv6 ネイバーテーブルは適切に入力されています。

```
<#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
```

アクション2:IPv6設定を確認します。

ファイアウォールの設定は、次のとおりです。

```
<#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
```

次のアップストリームデバイス設定により、設定ミスが分かります。

<#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
```

キャプチャ - 機能シナリオ :

サブネットマスクの変更 (/48 から /64 へ) により、問題が修正されています。機能シナリオで

の CAPI キャプチャの内容は、次のとおりです。

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 15:17:24.674403	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)

重要なポイント：

1. ルータは、アップストリームデバイスのMACアドレス(IP fc00:1:1:1::1)を要求するIPv6ネイバー送信要求メッセージを送信します。
2. ファイアウォールが IPv6 の Neighbor Advertisement で応答しています。
3. ルータが ICMP エコー要求を送信し、エコー応答を受信しています。

CAPO の内容：

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)

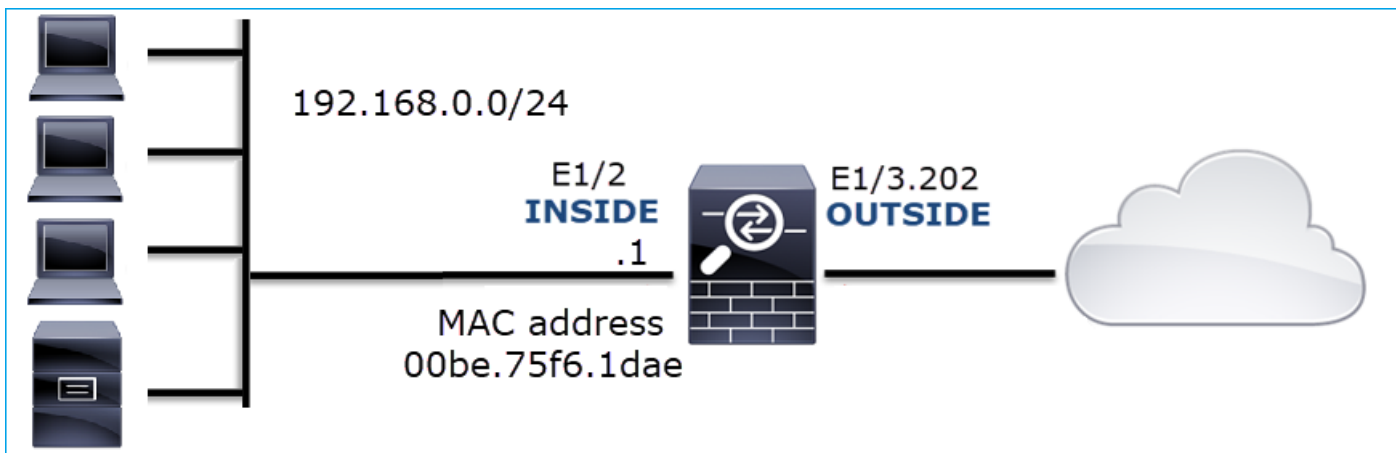
キーポイント：

1. ファイアウォールは、アップストリームデバイスのMACアドレス(IP fc00:1:1:2::2)を要求するIPv6ネイバー送信要求メッセージを送信します。
2. ファイアウォールが IPv6 の Neighbor Advertisement で応答しています。
3. ファイアウォールが ICMP エコー要求を送信しています。
4. ルータは、ダウンストリームデバイスのMACアドレス(IP fc00:1:1:1::1)を要求するIPv6ネイバー送信要求メッセージを送信します。
5. ファイアウォールが IPv6 の Neighbor Advertisement で応答しています。
6. ファイアウォールが ICMP エコー要求を送信し、エコー応答を受信しています。

Case 12.断続的な接続の問題 (ARPポイズニング)

問題の説明：内部ホスト(192.168.0.x/24)に、同じサブネット内のホストとの断続的な接続の問題がある

次の図は、このトポロジを示しています。



影響を受けるフロー：

送信元IP:192.168.0.x/24

宛先IP:192.168.0.x/24

プロトコル：任意

内部ホストの ARP キャッシュでポイズニングが発生していると見られます。

```

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>

```

キャプチャ分析

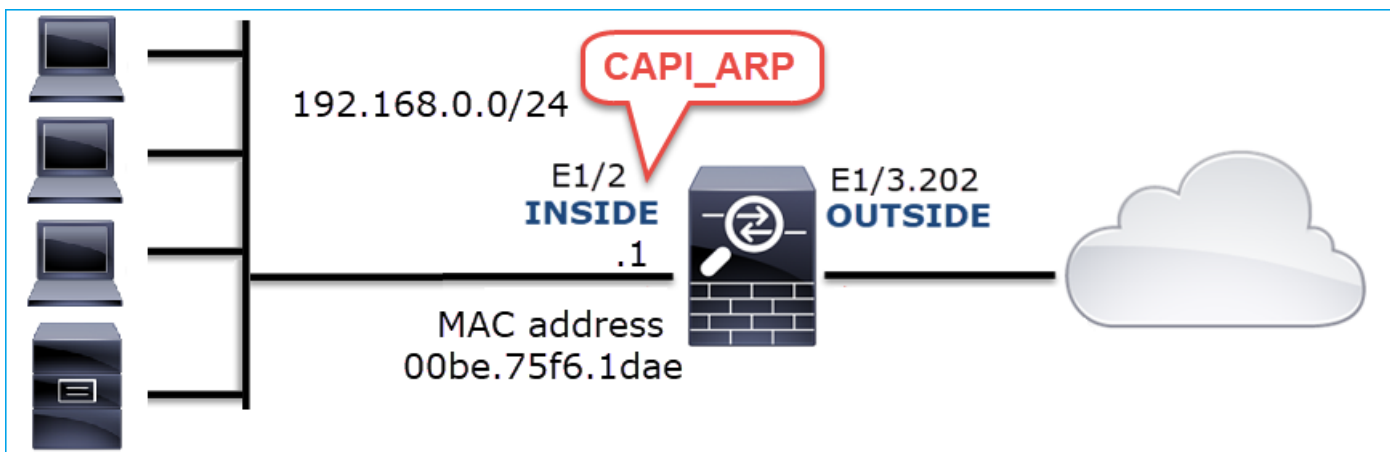
FTD LINA エンジンでのキャプチャを有効にします。

次のキャプチャでは、INSIDE インターフェイスの ARP パケットのみがキャプチャされます。

<#root>

firepower#

capture CAPI_ARP interface INSIDE ethernet-type arp



キャプチャ - 非機能シナリオ :

ファイアウォールの INSIDE インターフェイスでのキャプチャには、次の内容が含まれています

。

No.	Time	Source	Destination	Protocol	Length	Info
4	2019-10-25 10:01:55.179571	Vmware_2c:9b:a7	Broadcast	ARP	1	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	2	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	1	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	2	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	1	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	2	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	1	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	2	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	1	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	2	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	1	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	2	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	1	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	2	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	1	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	2	42 192.168.0.30 is at 00:be:75:f6:1d:ae

キー ポイント :

1. ファイアウォールが 192.168.0.x/24 ネットワーク内の IP に関するさまざまな ARP 要求を受信しています。
2. ファイアウォールが、それらすべて (プロキシ ARP) に独自の MAC アドレスで応答しています。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:NAT設定をチェックします。

NAT設定に関して、no-proxy-arpキーワードが以前の動作を妨げる可能性がある場合があります

。

<#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
```

アクション2:ファイアウォールインターフェイスでproxy-arp機能を無効にします。

「no-proxy-arp」キーワードを使用しても問題が解決しない場合は、インターフェイス自体でプロキシARPを無効にしてみてください。FTDの場合は、このドキュメントの作成時点で、FlexConfigを使用してコマンドを展開する必要があります (適切なインターフェイス名を指定します)。

```
sysopt noproxyarp INSIDE
```

Case 13.CPU Hogを引き起こすSNMPオブジェクトID(OID)の特定

このケースでは、SNMPバージョン3 (SNMPv3) パケットのキャプチャの分析に基づいて、メモリポーリングの特定のSNMP OIDをCPU占有 (パフォーマンスの問題) の根本原因として特定する方法を説明します。

問題の説明: データインターフェイスのオーバーランは継続的に増加します。さらに調査を進めると、インターフェイスオーバーランの根本原因であるCPUホグ (SNMPプロセスによって引き起こされる) も存在することが判明しました。

トラブルシューティングプロセスの次のステップは、SNMPプロセスによって引き起こされるCPU占有の根本原因を特定すること、特に、問題の範囲を絞り込んで、ポーリング時に潜在的にCPU占有を発生させる可能性があるSNMPオブジェクト識別子 (OID) を特定することでした。

現在、FTD LINA エンジンでは、ポーリングされているSNMP OIDをリアルタイムで確認するための「show」コマンドは提供されていません。

ポーリング用のSNMP OIDのリストはSNMPモニタリングツールから取得できますが、この場合は次の予防要因があります。

- FTD 管理者が SNMP モニタリングツールにアクセスできませんでした。
- プライバシーのために、FTD で SNMP バージョン 3 が認証とデータ暗号化によって設定されました。

キャプチャ分析

FTD 管理者は、SNMP バージョン 3 の認証とデータ暗号化のログイン情報を持っていたため、次

のアクションプランが提案されました。

1. SNMP パケットのキャプチャを取得します。
2. キャプチャを保存し、Wireshark の SNMP プロトコル設定を使用して、SNMP バージョン 3 パケットを復号するための SNMP バージョン 3 のログイン情報を指定します。復号されたキャプチャは、SNMP OID の分析と取得に使用されます。

snmp-server ホスト設定で使用するインターフェイスでの SNMP パケットのキャプチャを設定します。

```
<#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	0.326	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_

```

キーポイント：

1. SNMP 送信元および宛先のアドレス/ポートです。
2. privKey が Wireshark に認識されていないため、SNMP プロトコル PDU を復号できませんでした。
3. encryptedPDU プリミティブの値です。

推奨される対処法

このセクションに示されているアクションは、問題を絞り込むことを目的としています。

アクション1:SNMPキャプチャを復号化します。

キャプチャを保存し、Wireshark の SNMP プロトコル設定を編集して、パケットを復号するための SNMP バージョン 3 のログイン情報を指定します。

```
<#root>
```

```
firepower#
```

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

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

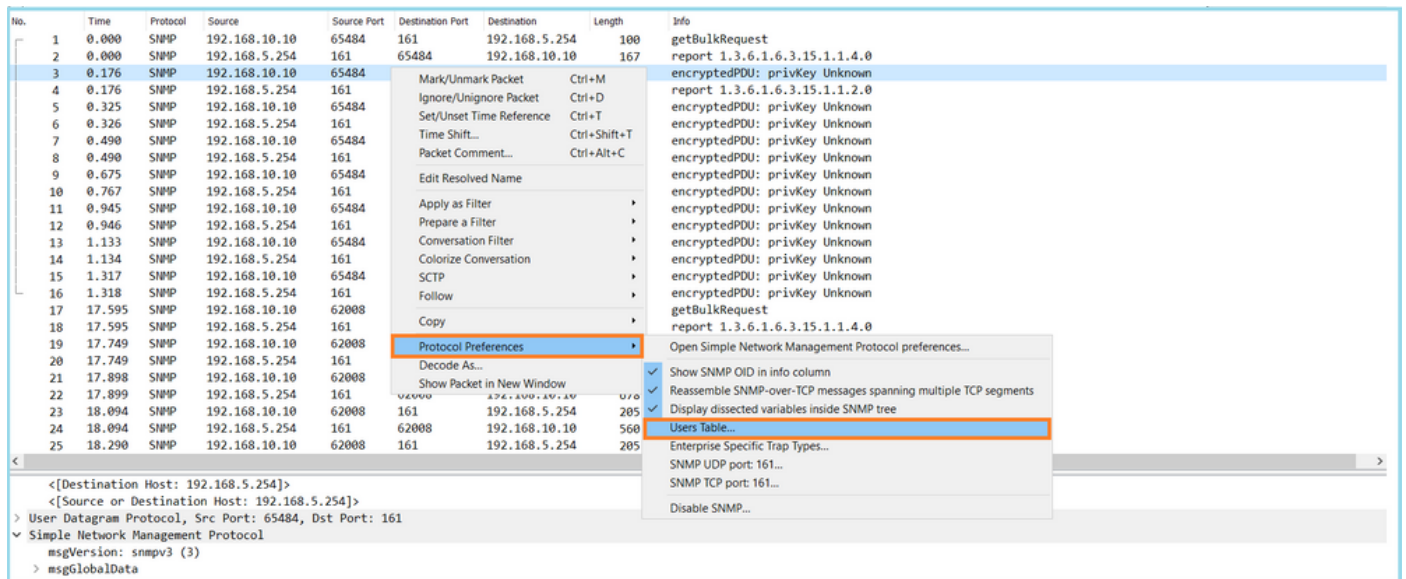
```
Address or name of remote host []? 192.168.10.253
```

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

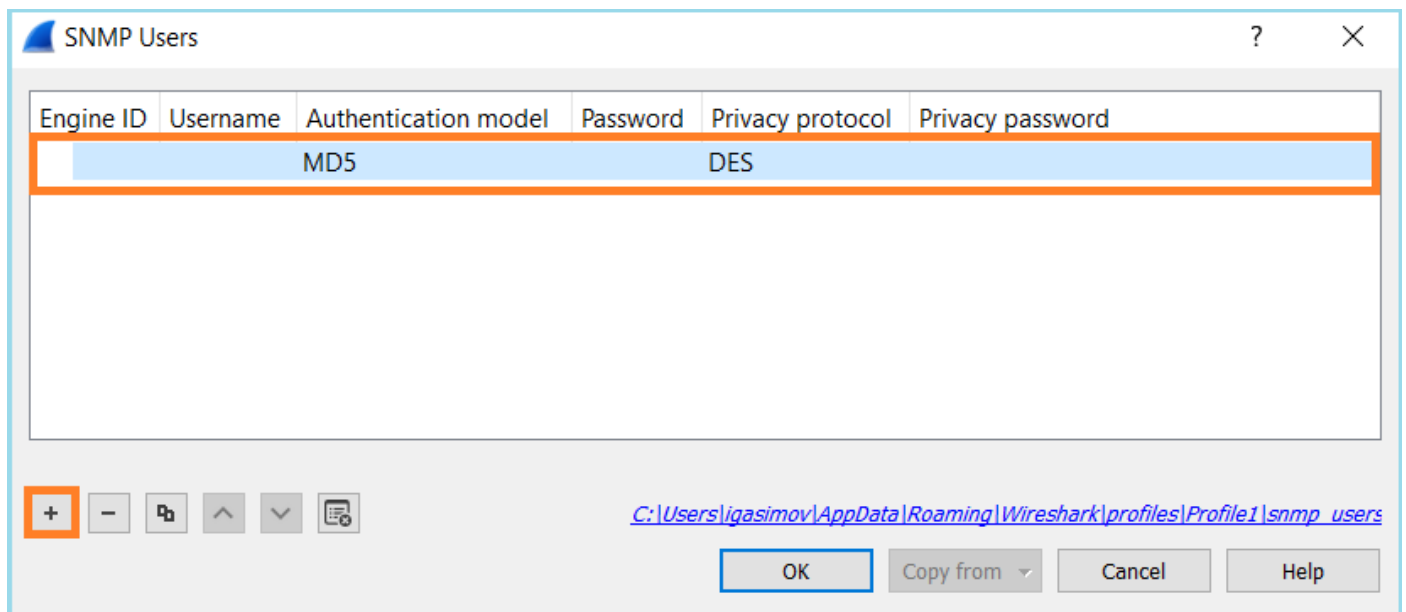
```
!!!!!!
```

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

図のように、Wireshark でキャプチャファイルを開き、SNMP パケットを選択して、[プロトコル設定 (Protocol Preferences)] > [ユーザーテーブル (Users Table)] に移動します。



SNMP ユーザーテーブルで、SNMP バージョン 3 のユーザー名、認証モデル、認証パスワード、プライバシープロトコル、およびプライバシーパスワードが指定されています (次の図には実際のログイン情報は表示されていません)。



SNMP ユーザー設定が適用されると、Wireshark に復号された SNMP PDU が表示されます。

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 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response 1.3.6.1.4.1.9.9.221.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.7.1.2 1.3.6.1.4.1.9.9.221.1.1
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.8.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response 1.3.6.1.4.1.9.9.221.1.1.1.17.1.1 1.3.6.1.4.1.9.9.221.1.1.1.17.1.2 1.3.6.1.4.1.9.9.221.1.1
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.18.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	get-response 1.3.6.1.4.1.9.9.221.1.1.1.19.1.1 1.3.6.1.4.1.9.9.221.1.1.1.19.1.2 1.3.6.1.4.1.9.9.221.1.1
15	1.317	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8
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
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 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	get-response 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8


```

msgData: encryptedPDU (1)
  encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703...
    Decrypted ScopedPDU: 303b04198000009fe1c6dad4930a00ef1fec2301621a4158bf1f40...
      contextEngineID: 8000009fe1c6dad4930a00ef1fec2301621a4158bf1f40...
      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)
  
```

キーポイント：

1. SNMP モニタリングツールが、SNMP getBulkRequest を使用して、親 OID 1.3.6.1.4.1.9.9.221.1 および関連する OID を照会し、処理しています。
2. FTDは、1.3.6.1.4.1.9.9.221.1に関連するOIDを含むget-responseで各getBulkRequestに回答しました。

アクション2:SNMP OIDを特定します。

[次の図のように、SNMP Object Navigator](#) には、OID 1.3.6.1.4.1.9.9.221.1 が CISCO-ENHANCED-MEMPOOL-MIB という名前の Management Information Base (MIB) に属していることが示されています。

Tools & Resources
SNMP Object Navigator

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

Support Case Manager
 Cisco Community
 MIB Locator

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	Download	Download	View Dependencies
2. SNMPv2-TC	Download	Download	View Dependencies
3. SNMPv2-CONF	Not Required	Download	View Dependencies
4. SNMP-FRAMEWORK-MIB	Download	Download	View Dependencies
5. CISCO-SMI	Download	Download	View Dependencies
6. ENTITY-MIB	Download	Download	View Dependencies
7. HCNUM-TC	Download	Download	View Dependencies
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	Download	Download	

2. Wireshark の [編集 (Edit)] > [設定 (Preferences)] > [名前解決 (Name Resolution)] ウィンドウでは、[OID解決を有効にする (Enable OID Resolution)] がオンになっています。[SMI (MIBおよびPIBパス) (SMI (MIB and PIB modules))] ウィンドウで、ダウンロードした MIB を含むフォルダと SMI (MIB および PIB モジュール) を指定します。CISCO-ENHANCED-MEMPOOL-MIB は、モジュールのリストに自動的に追加されます。

The screenshot shows the Wireshark interface with the 'Name Resolution' and 'SMI Modules' dialog boxes open. In the 'Name Resolution' dialog, the 'Enable OID resolution' checkbox is checked. In the 'SMI Modules' dialog, the directory path 'C:/Users/Administrator/Downloads/SNMPMIBS' is entered, and 'CISCO-ENHANCED-MEMPOOL-MIB' is selected in the module list.

3. Wireshark が再起動すると、OID 解決がアクティブになります。

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::smStatsUnknownEngineIDs.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMIBObjects
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report SNMP-USER-BASED-SM-MIB::smStatsNotInTimeWindows.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMIBObjects
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolType.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolType
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolAlternate.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPool
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolValid.1.8
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsed.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsed
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolFree.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsedOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPool
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsed.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	600	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsedOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPool

▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.1) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: System memory
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.2) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: System memory
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.3) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_MSGLYR
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.4) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_HEAPCACHE_1
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.5) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_HEAPCACHE_0
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.6) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_DMA_ALT1
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.7) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_DMA
▼ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.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::compMemPoolName.1.8) CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_GLOBAL_SHARED

キャプチャファイルの復号された出力に基づいて、SNMP モニタリングツールは、FTD のメモリプールの使用率に関するデータを定期的に (10 秒間隔で) ポーリングしています。TechNoteの記事「[メモリ関連統計情報のASA SNMPポーリング](#)」で説明されているように、SNMPを使用してグローバル共有プール(GSP)の使用率をポーリングすると、CPUの使用率が高くなります。この場合、キャプチャから、グローバル共有プールの使用率が SNMP getBulkRequest プリミティブの一部として定期的にポーリングされていたことが分かります。

SNMP プロセスによって発生する CPU 占有を最小限に抑えるために、記事に記載されている SNMP に関する CPU 占有の軽減手順に従い、GSP に関連する OID をポーリングしないことが推奨されています。GSP に関連する OID の SNMP ポーリングがない場合、SNMP プロセスによって発生する CPU 占有は見られず、オーバーランのレートは大幅に減少しています。

関連情報

- [Cisco Firepower Management Center のコンフィギュレーションガイド](#)
- [Firepower Threat Defense アクセス コントロール ポリシー ルール アクションの明確化](#)
- 『[Work with Firepower Threat Defense Captures and Packet Tracer](#)』
- 『[Learn Wireshark](#)』

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。