分析 Firepower 防火牆擷取，以有效針對網路問題進行疑難排解

目錄

簡介

必要條件

需求

採用元件

背景資訊

如何收集和匯出NGFW產品系列中的捕獲？

收集FXOS捕獲

啟用和收集FTD Lina擷取

啟用和收集FTD Snort擷取

疑難排解

案例1.輸出介面上沒有TCP SYN

捕獲分析

建議的操作

可能的原因和建議的操作摘要

案例2.來自客戶端的TCP SYN，來自伺服器的TCP RST

捕獲分析

建議的操作

案例3.來自一個終端的TCP三次握手+ RST

捕獲分析

3.1 — 客戶端的TCP三次握手+延遲RST

建議的操作

3.2 - TCP三次握手+來自客戶端的延遲FIN/ACK +來自伺服器的延遲RST

建議的操作

3.3 — 客戶端的TCP三次握手+延遲RST

建議的操作

3.4 — 來自伺服器的TCP三次握手+即時RST

建議的操作

案例4.來自使用者端的TCP RST

捕獲分析

建議的操作

案例5.TCP傳輸緩慢（場景1）

案例 1.傳輸緩慢

捕獲分析

建議的操作

案例 2.快速傳輸

案例6.TCP傳輸緩慢（案例2）

捕獲分析

建議的操作

案例7.TCP連線問題（封包損毀）

捕獲分析

建議的操作

案例8.UDP連線問題（缺少資料包）
- 捕獲分析
- 建議的操作

案例9.HTTPS連線問題（場景1）
- 捕獲分析
- 建議的操作

案例10.HTTPS連線問題（場景2）
- 捕獲分析
- 建議的操作

案例11.IPv6連線問題
- 捕獲分析
- 建議的操作

案例12.間歇性連線問題（ARP中毒）
- 捕獲分析
- 建議的操作

案例13.標識導致CPU佔用的SNMP對象識別符號(OID)
- 捕獲分析
- 建議的操作

相關資訊

簡介

本文件說明各種封包擷取分析技術，旨在有效對網路問題進行疑難排解。

必要條件

需求

思科建議您瞭解以下主題：

- Firepower平台架構
- NGFW日誌
- NGFW Packet Tracer

此外，開始分析資料包捕獲之前，強烈建議滿足以下要求：

- 瞭解協定操作 — 如果您不瞭解捕獲的協定如何運行，請不要開始檢查資料包捕獲。
- 瞭解拓撲 — 您必須瞭解端對端的傳輸裝置。如果這不可能，您至少必須知道上游和下游裝置。
- 瞭解設備 — 您必須瞭解裝置如何處理資料包、涉及的介面（入口/出口）、裝置架構是什麼，以及各種捕獲點。
- 瞭解組態 — 您必須知道裝置應該如何根據以下條件處理封包流：
  - 路由/輸出介面
  - 應用的策略
  - 網路位址轉譯(NAT)
- 瞭解可用工具 — 除了捕獲之外，建議準備好應用其他工具和技術（如日誌記錄和跟蹤程式），並在需要時將其與捕獲的資料包相關聯

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- 大多數場景基於運行FTD軟體6.5.x的FP4140。
- FMC運行軟體6.5.x。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

資料包捕獲是當今最被忽視的故障排除工具之一。每天，Cisco TAC都可在分析捕獲的資料時解決許多問題。

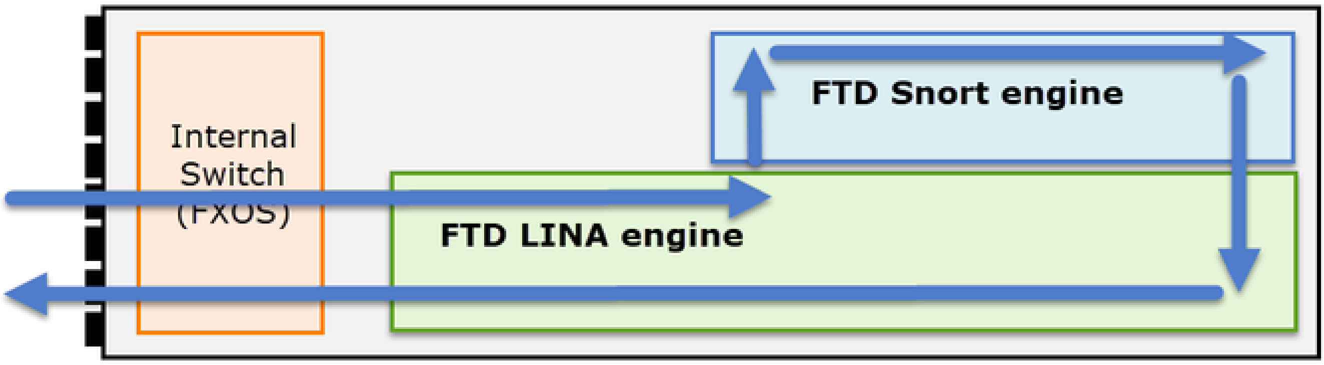
本文檔的目標是幫助網路和安全工程師主要基於資料包捕獲分析來識別和排除常見網路問題。

本文提供的所有情境均基於思科技術協助中心(TAC)中可見的實際使用者案例。

本檔案從思科新世代防火牆(NGFW)的角度介紹封包擷取，但相同的概念同樣適用於其他裝置型別。

如何收集和匯出NGFW產品系列中的捕獲？

若是Firepower裝置(1xxx、21xx、41xx、93xx)和Firepower威脅防禦(FTD)應用程式，資料包處理視覺化，如下圖所示。



1. 封包進入輸入介面，並由機箱內部交換器處理。
2. 封包進入FTD Lina引擎，主要執行L3/L4檢查。
3. 如果策略要求封包由Snort引擎檢查（主要是L7檢查）。
4. Snort引擎傳回封包的判定結果。
5. LINA 引擎根據 Snort 的判定結果捨棄或轉送封包.
6. 封包透過內部機箱交換器離開機箱。

根據所示架構，FTD擷取可以在三(3)個不同地方進行：

- FXOS

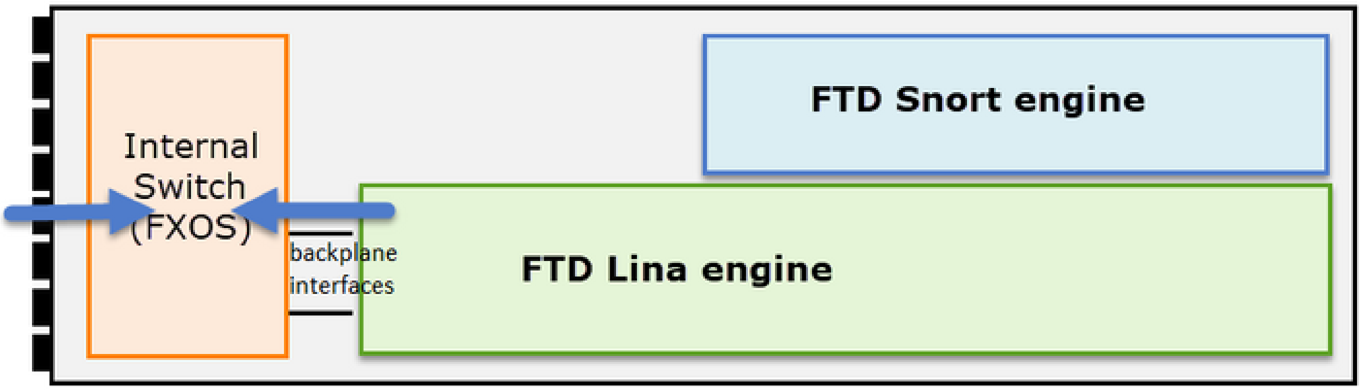
- FTD Lina引擎
- FTD Snort引擎

收集FXOS捕獲

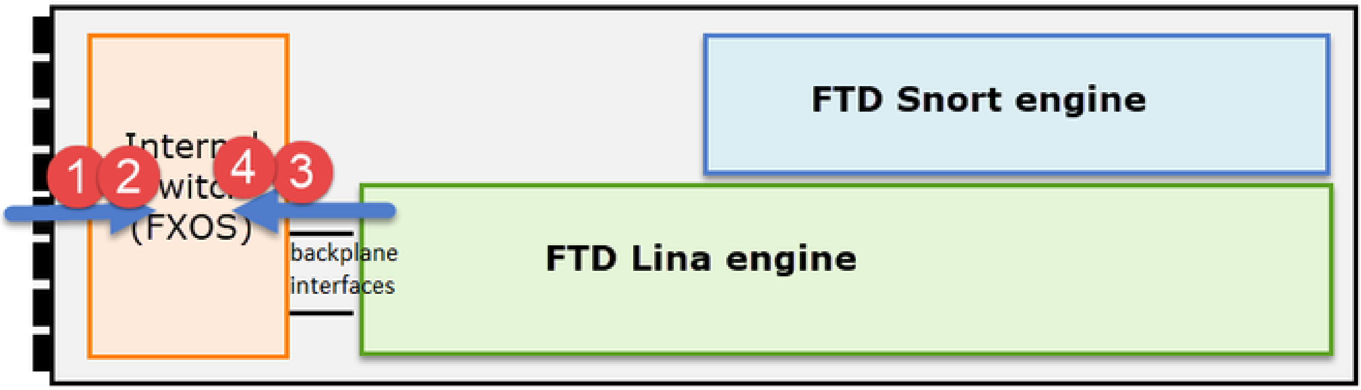
本檔案將說明此程式：

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

FXOS擷取只能從內部交換器視點沿輸入方向擷取，如下圖所示。



此處顯示，每個方向有兩個擷取點（由於內部交換器架構）。



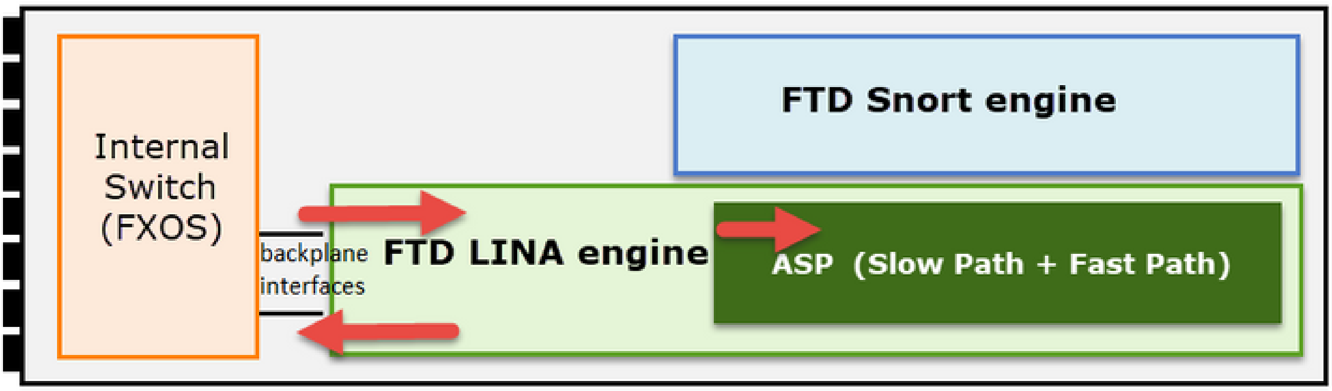
點2、3和4中捕獲的資料包具有虛擬網路標籤(VNTag)。

註:FXOS機箱級捕獲僅在FP41xx和FP93xx平台上可用。FP1xxx和FP21xx不提供此功能。

啟用和收集FTD Lina擷取

主要捕獲點：

- 輸入介面
- 輸出介面
- 加速安全路徑(ASP)



您可以使用Firepower管理中心使用者介面(FMC UI)或FTD CLI啟用和收集FTD Lina捕獲。

在INSIDE介面上從CLI啟用捕獲：

<#root>

```
firepower#

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

此捕獲匹配IP地址192.168.103.1和192.168.101.1之間的雙向流量。

啟用ASP捕獲以檢視FTD Lina引擎丟棄的所有資料包：

<#root>

```
firepower#

capture ASP type asp-drop all
```

將FTD Lina擷取匯出至FTP伺服器：

<#root>

```
firepower#

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

將FTD Lina擷取匯出至TFTP伺服器：

<#root>

```
firepower#

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

自FMC 6.2.x版本起，您可以從FMC UI啟用和收集FTD Lina擷取。

從FMC管理的防火牆收集FTD擷取的另一種方法如下。

步驟 1

在LINA或ASP擷取的情況下，將擷取複製到FTD磁碟。

<#root>

```
firepower#

copy /pcap capture:capin disk0:capin.pcap


Source capture name [capin]?

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

步驟 2

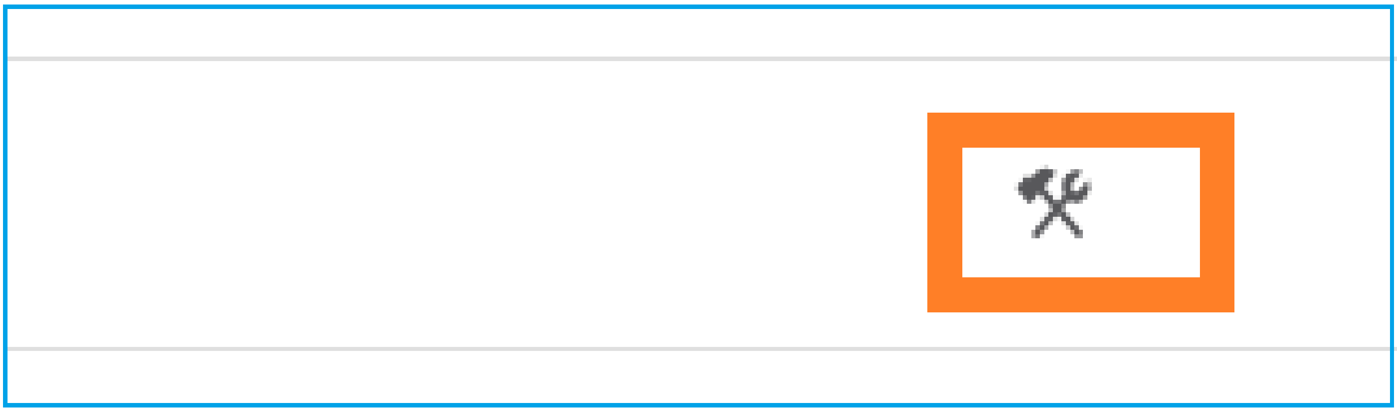
導航到專家模式，找到儲存的捕獲，並將其複製到/ngfw/var/common位置：

<#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
```

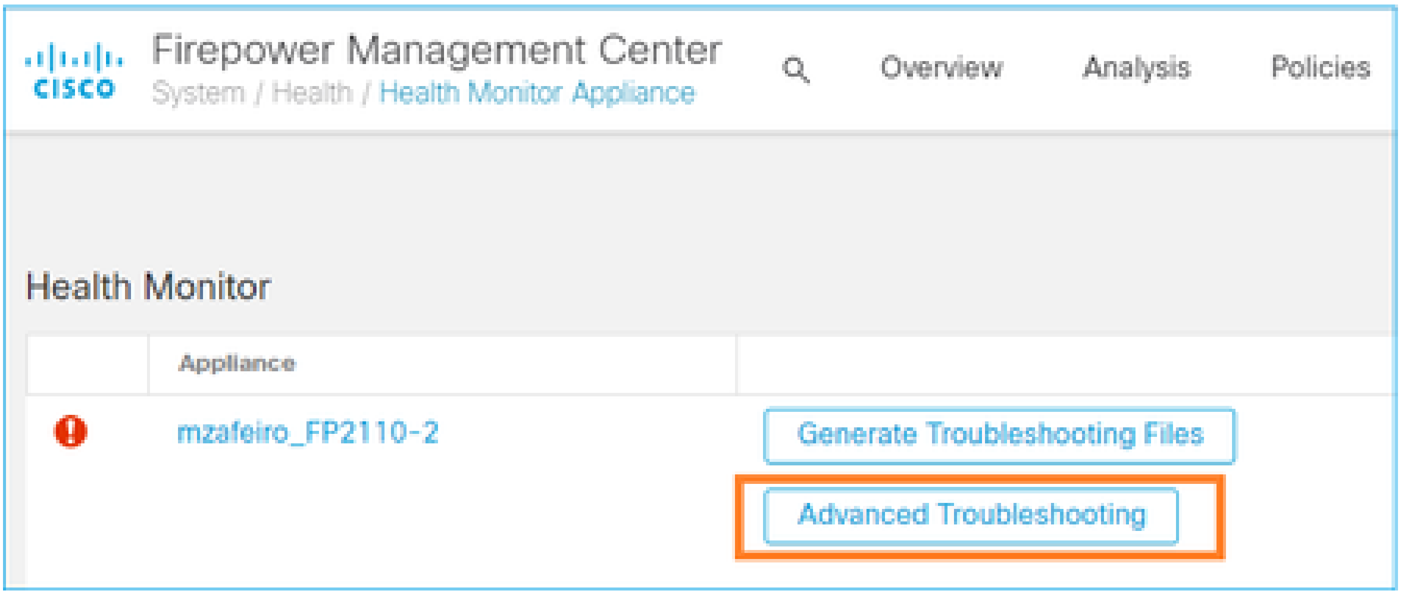
步驟 3

登入管理FTD的FMC，然後導覽至Devices > Device Management。找到FTD裝置，然後選擇Troubleshoot圖示：

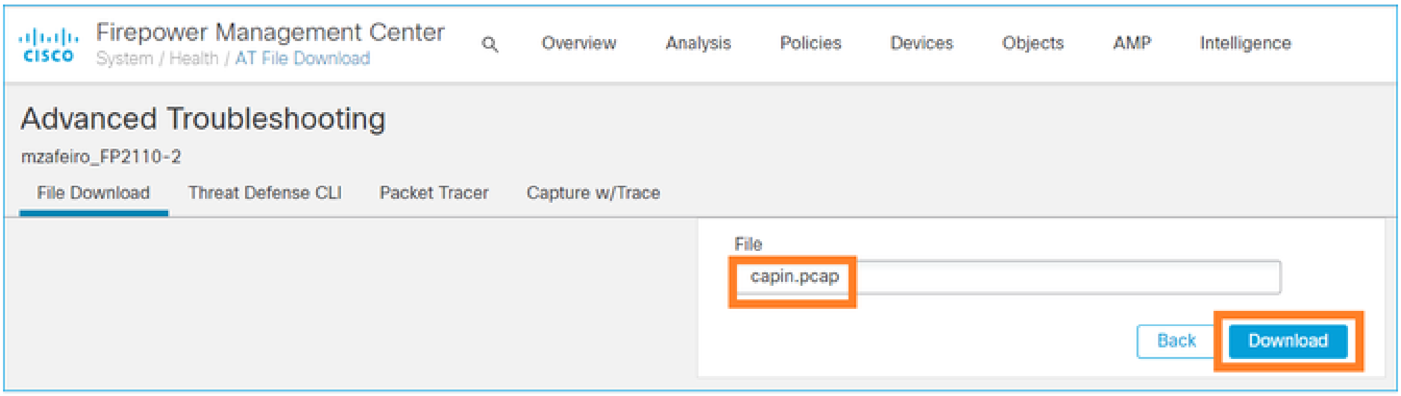


步驟 4

選擇Advanced Troubleshooting:


Firepower Management Center
System / Health / Health Monitor Appliance
Overview
Analysis
Policies
Health Monitor
Appliance
mzafeiro_FP2110-2
Generate Troubleshooting Files
Advanced Troubleshooting

指定捕獲檔名並選擇 下載：



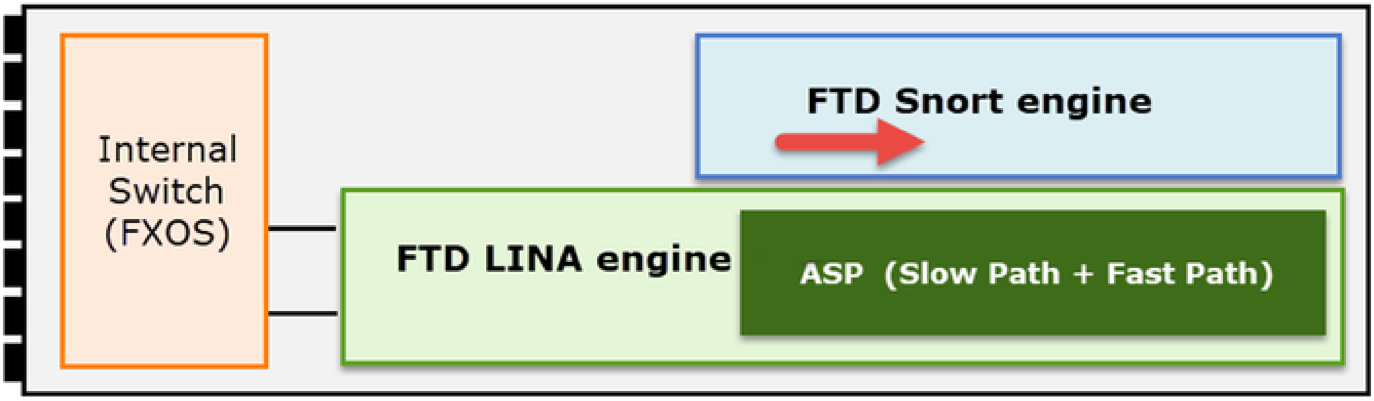
File: capin.pcap

有關如何從FMC UI啟用/收集捕獲的更多示例，請查閱以下文檔：

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

啟用和收集FTD Snort擷取

捕獲點顯示在此處的影象中。



啟用Snort級別捕獲：

```
<#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
```

將捕獲寫入名為capture.pcap的檔案並通過FTP複製到遠端伺服器：

```
<#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.

>
```

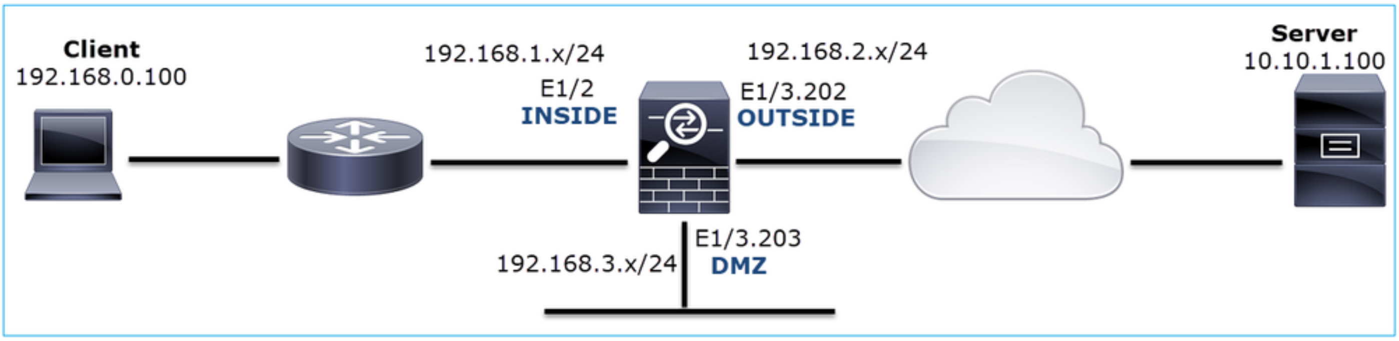
有關包含不同捕獲過濾器的更多Snort級別捕獲示例，請查閱以下文檔：

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

疑難排解

案例1.輸出介面上沒有TCP SYN

拓撲如下圖所示：



問題說明：HTTP無法正常工作

受影響的流：

源IP:192.168.0.100

Dst IP:10.10.1.100

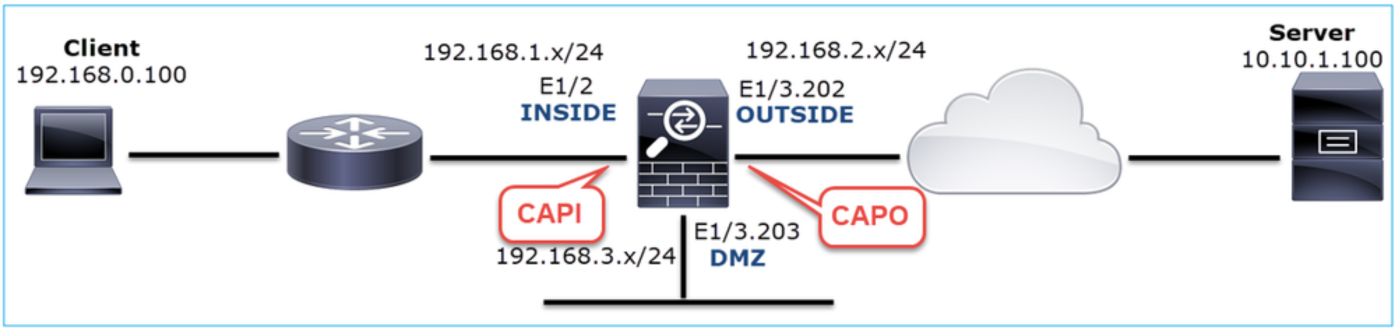
協定：TCP 80

捕獲分析

在FTD LINA引擎上啟用擷取：

<#root>

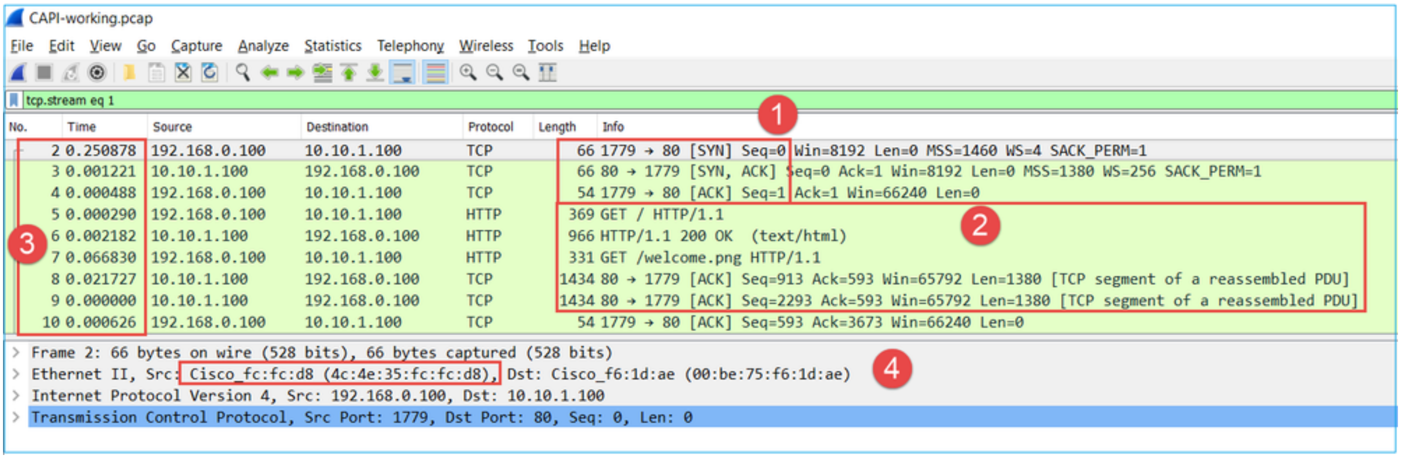
```
firepower#

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
```



捕獲 — 功能場景：

作為基準，從功能場景中捕獲資料始終非常有用。

在NGFW INSIDE介面上進行的捕獲，如下圖所示：


CAPI-working.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
1
2
3
4

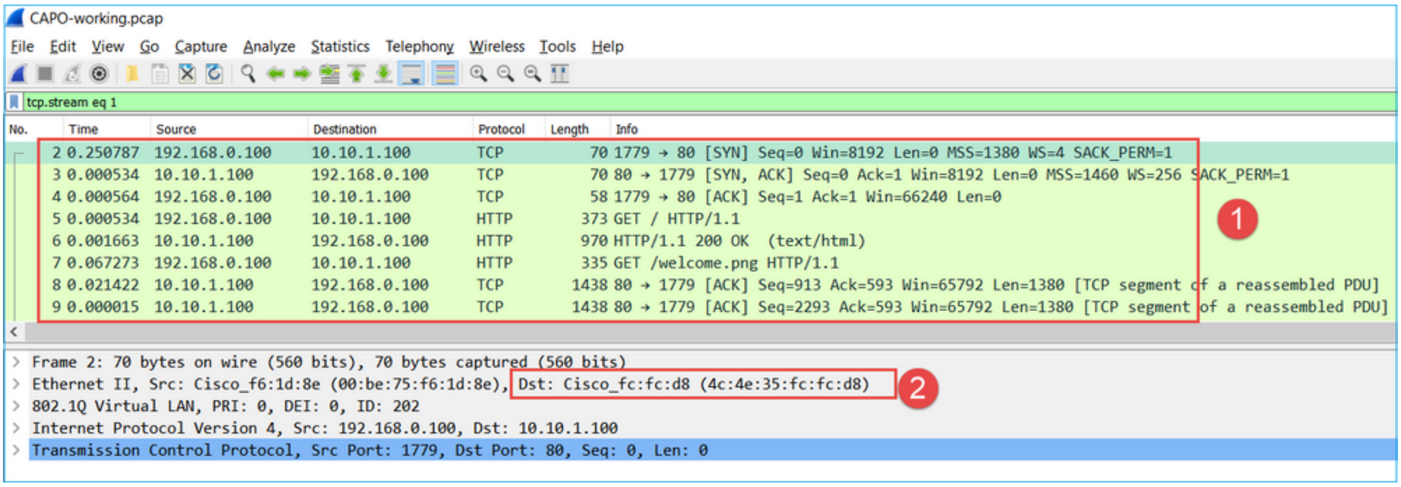
tcp.stream eq 1

```
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: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
```

重點：

1. TCP三次握手。
2. 雙向資料交換。
3. 資料包之間無延遲（基於資料包之間的時間差）
4. 源MAC是正確的下游裝置。

在NGFW OUTSIDE介面上進行的捕獲，如下圖所示：


CAPO-working.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
1
2

tcp.stream eq 1

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_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: 1779, Dst Port: 80, Seq: 0, Len: 0
```

重點：

1. 與CAPI捕獲中的資料相同。
2. 目標MAC是正確的上游裝置。

捕獲 — 非功能方案

從裝置CLI中，捕獲如下所示：

<#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內容：

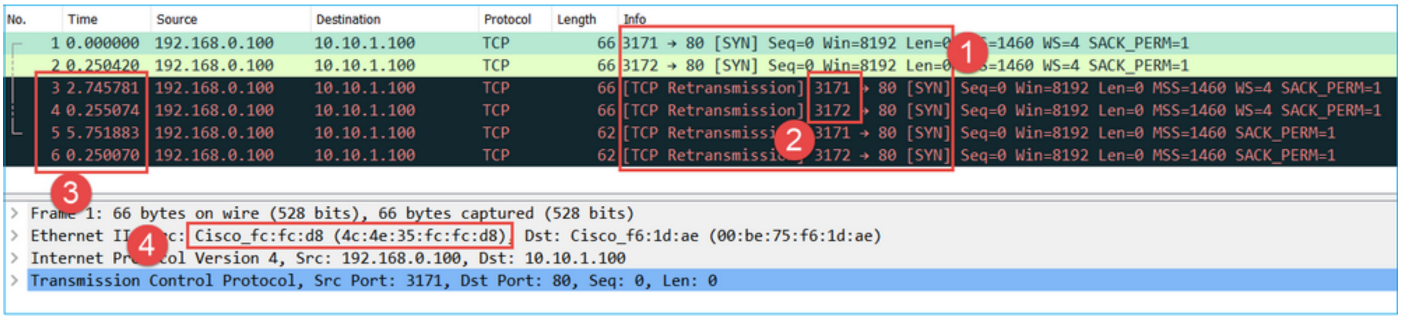
<#root>

```
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
```

這是CAPI捕獲在Wireshark中的影象：



```
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
```

重點：

1. 只看到TCP SYN封包（無TCP三次握手）。
2. 2個TCP會話（源埠3171和3172）無法建立。來源使用者端會重新傳送TCP SYN封包。這些重新傳輸的資料包由Wireshark識別為TCP重新傳輸。
3. TCP重新傳輸每~3秒、每6秒進行一次。
4. 源MAC地址來自正確的下游裝置。

根據2個擷取可得出以下結論：

- 特定5元組(src/dst IP、src/dst port、protocol)的資料包到達預期介面(INSIDE)上的防火牆。
- 封包不會離開預期介面(OUTSIDE)上的防火牆。

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.檢查模擬資料包的跟蹤。

使用Packet Tracer工具檢視防火牆應如何處理資料包。如果防火牆訪問策略丟棄了資料包，則模擬資料包的跟蹤看起來與以下輸出類似：

<#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
```

行動2.檢查活動資料包的蹤跡。

啟用資料包跟蹤檢查防火牆如何處理實際TCP SYN資料包。預設情況下，僅追蹤前50個輸入封包：

<#root>

```
firepower#

capture CAPI trace
```

清除擷取緩衝區：

<#root>

```
firepower#

clear capture /all
```

如果封包被防火牆存取原則捨棄，追蹤軌跡會與以下輸出類似：

<#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
```

行動3.檢查FTD Lina記錄。

若要透過FMC在FTD上設定系統日誌，請參閱以下檔案：

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

強烈建議為FTD Lina記錄設定外部系統日誌伺服器。如果沒有配置遠端系統日誌伺服器，請在進行故障排除時在防火牆上啟用本地緩衝區日誌。本示例中顯示的日誌配置是一個良好的起點：

```
<#root>

firepower#

show run logging

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

將終端尋呼機設定為24行，以便控制終端尋呼機：

```
<#root>

firepower#

terminal pager 24
```

清除擷取緩衝區：

```
<#root>

firepower#

clear logging buffer
```

測試連線並使用解析器過濾器檢查日誌。在此範例中，封包被防火牆存取原則捨棄：

```
<#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
```

行動4.檢查防火牆ASP丟棄。

如果您懷疑封包被防火牆捨棄，可以在軟體層級看到防火牆捨棄的所有封包的計數器：

```
<#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
```

您可以啟用捕獲以檢視所有ASP軟體級別的丟棄：

```
<#root>

firepower#

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

提示：如果您對資料包內容不感興趣，則只能捕獲資料包報頭（僅報頭選項）。這樣您就可以在擷取緩衝區中擷取更多封包。此外，還可以將捕獲緩衝區的大小（預設情況下為500KB）增加到最多32MB的值（緩衝區選項）。最後，從FTD版本6.3開始，檔案大小選項允許您配置高達10GB的捕獲檔案。在這種情況下，您只能看到採用pcap格式的捕獲內容。

若要檢查捕獲內容，可以使用篩選器縮小搜尋範圍：

```
<#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
```

在這種情況下，由於已在介面級別跟蹤資料包，因此ASP捕獲中不會提及丟棄的原因。請記住，只能在一個位置追蹤封包（輸入介面或ASP捨棄）。在這種情況下，建議使用多個ASP丟棄並設定特定ASP丟棄原因。以下是建議的方法：

1.清除當前ASP刪除計數器：

```
<#root>

firepower#

clear asp drop
```

2.通過防火牆傳送故障排除的流（運行測試）。

3.再次檢查ASP下拉計數器並記下增加的。

```
<#root>

firepower#

show asp drop

Frame drop:
  No route to host (

no-route

)                                                   234
  Flow is denied by configured rule (

acl-drop

)                                 71
```

4.為出現的特定丟包啟用ASP捕獲：

```
<#root>
```

```
firepower#

capture ASP_NO_ROUTE type asp-drop no-route

firepower#

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

5.通過防火牆傳送您進行故障排除的流（運行測試）。

6.檢查ASP捕獲。在這種情況下，由於缺少路由，資料包被丟棄：

```
<#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
```

行動5.檢查FTD Lina連線表。

有時您預計資料包會輸出介面「X」，但無論出於什麼原因，它都會輸出介面「Y」。防火牆輸出介面判斷取決於以下操作順序：

1. 已建立的連線查詢
2. 網路地址轉換(NAT)查詢 — UN-NAT（目標NAT）階段優先於PBR和路由查詢。
3. 原則型路由(PBR)
4. 路由表查詢

檢查FTD連線表：

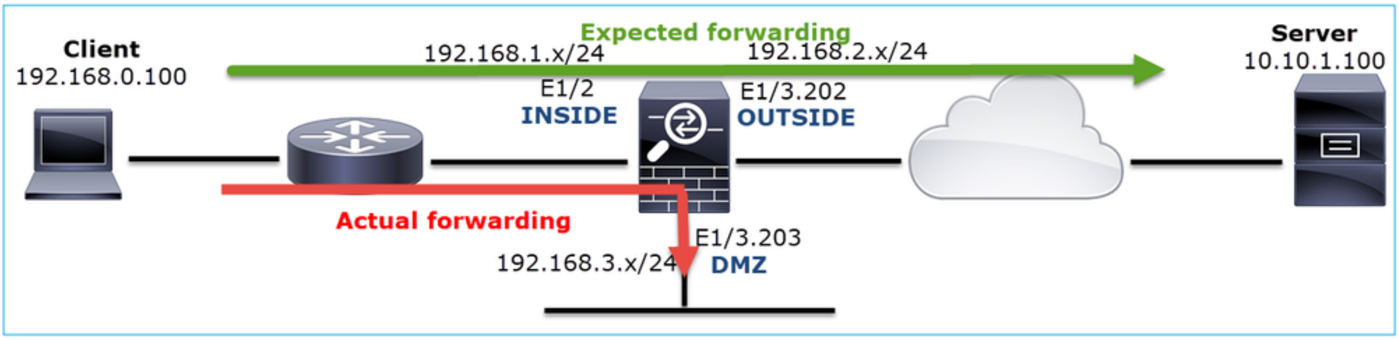
```
<#root>

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
```

重點：

- 根據標誌(Aa)，連線處於初始狀態（半開啟 — 防火牆只看到TCP SYN）。
- 根據來源/目的地連線埠，輸入介面為INSIDE，輸出介面為DMZ。

您可以在此處的影象中直觀顯示它：



註：由於所有FTD介面的安全等級都是0，因此show conn輸出中的介面順序取決於介面編號。具體而言，具有更高vpif-num（虛擬平台介面編號）的介面被選為inside，而具有更低vpif-num的介面被選為outside。您可以使用show interface detail指令看到介面vpif值。相關增強功能，思科錯誤ID CSCvi15290

ENH:FTD顯示FTD 'show conn'輸出中的連線方向性

<#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
```

注意：從Firepower軟體版本6.5到ASA 9.13.x版本開始，show conn long和show conn detail命令輸出提供有關連線啟動器和響應器的資訊

輸出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
```

輸出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
```

此外，show conn long還會顯示NATed IPs（在網路地址轉換的情況下）：

<#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
```

行動6.檢查防火牆位址解析通訊協定(ARP)快取。

如果防火牆無法解析下一跳，防火牆會以靜默方式丟棄原始資料包（本例中為TCP SYN），並繼續傳送ARP請求，直到解析下一跳。

要檢視防火牆ARP快取，請使用命令：

<#root>

```
firepower#

show arp
```

此外，若要檢查是否有未解析的主機，可以使用命令：

<#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
```

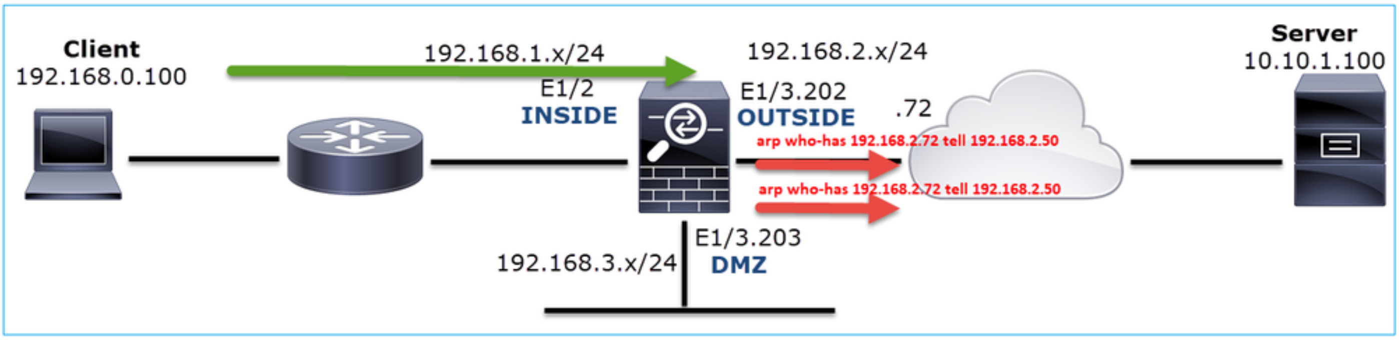
如果要進一步檢查ARP操作，可以啟用特定於ARP的捕獲：

<#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
```

在此輸出中，防火牆(192.168.2.50)嘗試解析下一躍點(192.168.2.72)，但沒有ARP應答



此處的輸出顯示了具有正確ARP解析的功能場景：

<#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
```

如果沒有ARP專案，則即時TCP SYN資料包的跟蹤會顯示：

<#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
```

從輸出中可看出，追蹤軌跡顯示Action: allow，即使下一個躍點無法連線且防火牆以靜默方式捨棄封包！在這種情況下，還必須檢查Packet Tracer工具，因為它提供了更精確的輸出：

<#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)/

在最新的ASA/Firepower版本中，以前的消息已最佳化為：

<#root>

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

Check ARP table (show arp) has entry for nexthop

```
., Drop-location: f
```

可能的原因和建議的操作摘要

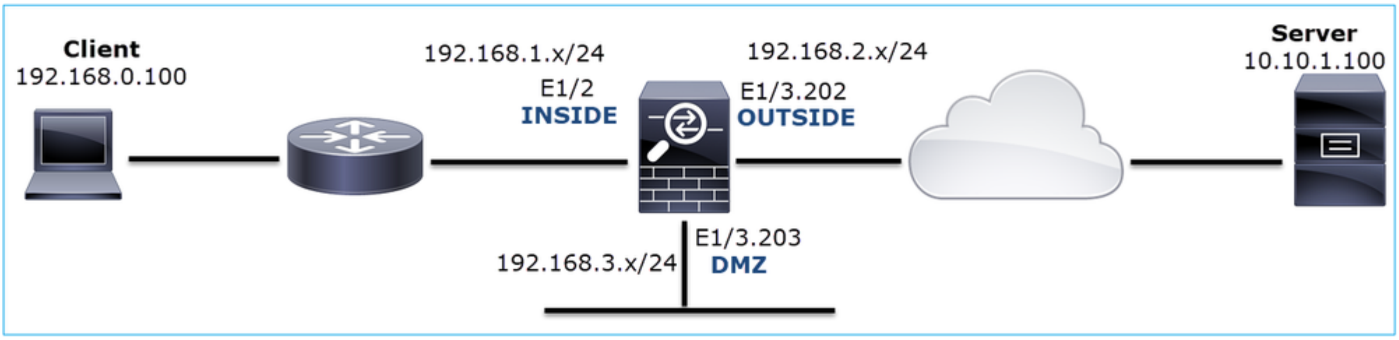
如果您在輸入介面上只看到TCP SYN封包，但沒有從預期的輸出介面發出任何TCP SYN封包，則一些可能的原因如下：

可能的原因	建議的操作
防火牆存取原則捨棄封包。	• 使用packet Tracer或capture w/trace檢視如何防火牆處理資料包。 • 檢查防火牆日誌。 • 檢查防火牆ASP丟棄(show asp drop或capture type asp-drop)。 • 檢查FMC連線事件。假設規則已啟用日誌記錄。
捕獲篩選器錯誤。	• 使用packet-tracer或capture w/trace檢視是否有修改源IP或目標IP的NAT轉換。在這種情況下，調整您的捕獲過濾器。 • show conn long命令輸出顯示NATed IP。
將封包傳送到不同的輸出介面。	• 使用packet Tracer或capture w/trace檢視防火牆如何處理資料包。記住有關輸出介面確定、當前連線、UN-NAT、PBR和路由表查詢的操作順序。 • 檢查防火牆日誌。 • 檢查防火牆連線表(show conn)。

	如果資料包由於與當前連線匹配而被傳送到錯誤的介面，請使用命令clear conn address 並指定要清除的連線的5元組。
沒有通往目的地的路由。	• 使用packet Tracer或capture w/trace檢視如何防火牆處理資料包。 • 檢查防火牆ASP丟棄(show asp drop)以獲取no-route drop原因。
輸出介面上沒有ARP專案。	• 檢查防火牆ARP快取(show arp)。 • 使用packet Tracer檢視是否有有效的鄰接關係。
輸出介面已關閉。	檢查防火牆上show interface ip brief命令的輸出，並驗證介面狀態。

案例2.來自客戶端的TCP SYN，來自伺服器的TCP RST

下圖顯示拓撲：



問題說明：HTTP無法正常工作

受影響的流：

源IP:192.168.0.100

Dst IP:10.10.1.100

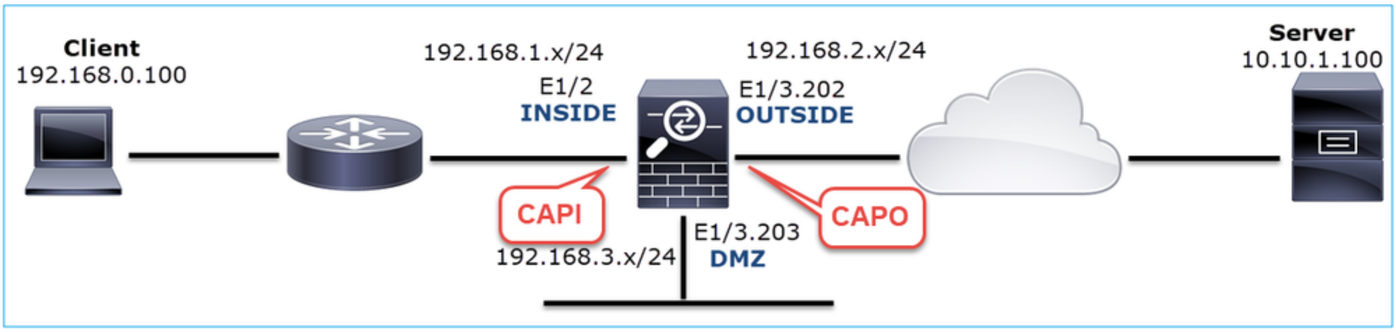
協定：TCP 80

捕獲分析

在FTD LINA引擎上啟用擷取。

<#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
```



捕獲 — 非功能場景：

從裝置CLI中捕獲如下所示：

<#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內容：

<#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內容：

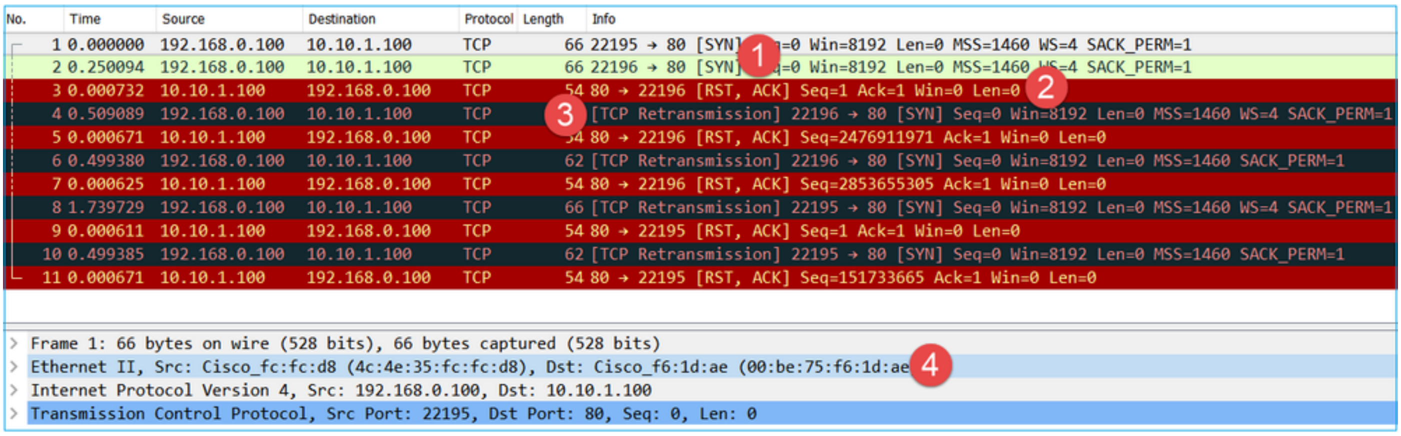
<#root>

```
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 P0 10.10.1.100.80 > 192.168.0.100.22196:

R

 0:0(0) ack 4785345 win 0
   4: 05:20:37.414269   802.1Q vlan#202 P0 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 P0 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 P0 192.168.0.100.22196 > 10.10.1.100.80:
```

s

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

此圖顯示CAPI在Wireshark中的捕獲。

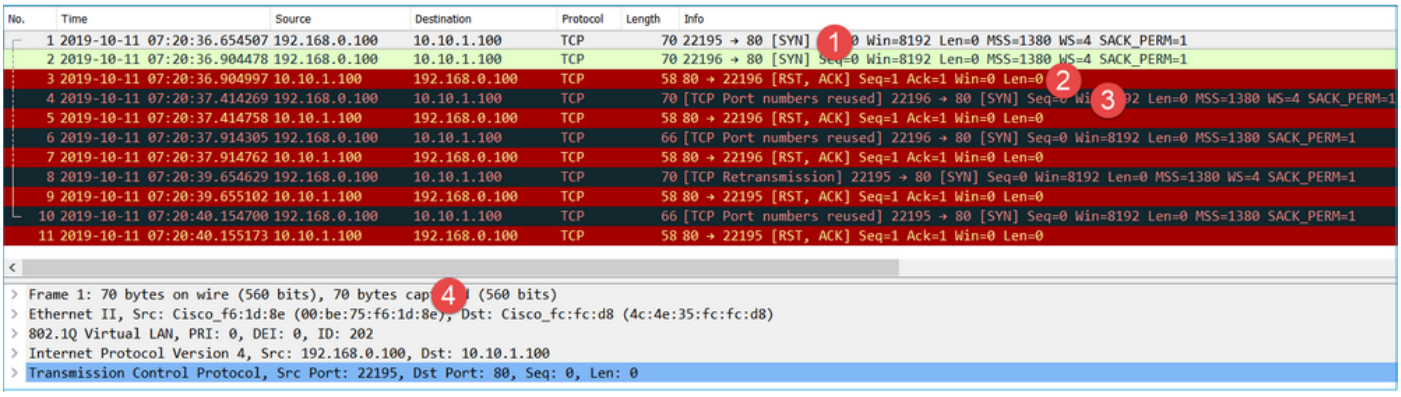


```
No.  Time      Source         Destination    Protocol Length Info
   1 0.000000  192.168.0.100  10.10.1.100    TCP      66 22195 → 80 [SYN] =0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
   2 0.250094  192.168.0.100  10.10.1.100    TCP      66 22196 → 80 [SYN] =0 Win=8192 Len=0 MSS=1460 =4 SACK_PERM=1
   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         [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
```

重點：

1. 來源傳送TCP SYN封包。
2. TCP RST會傳送到來源。
3. 來源重新傳輸TCP SYN封包。
4. MAC地址正確（在入口資料包上，源MAC地址屬於下游路由器，目的MAC地址屬於防火牆INSIDE介面）。

此圖顯示Wireshark中的CAPO捕獲：



```
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] 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= Wi 92 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 cap  (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: 22195, Dst Port: 80, Seq: 0, Len: 0
```

重點：

1. 來源傳送TCP SYN封包。
2. TCP RST到達外部介面。
3. 來源重新傳輸TCP SYN封包。
4. MAC地址正確（在出口資料包上，防火牆OUTSIDE是源MAC，上游路由器是目標MAC）。

根據2個擷取可得出以下結論：

- 客戶端和伺服器之間的TCP三次握手沒有完成
- 存在到達防火牆輸出介面的TCP RST

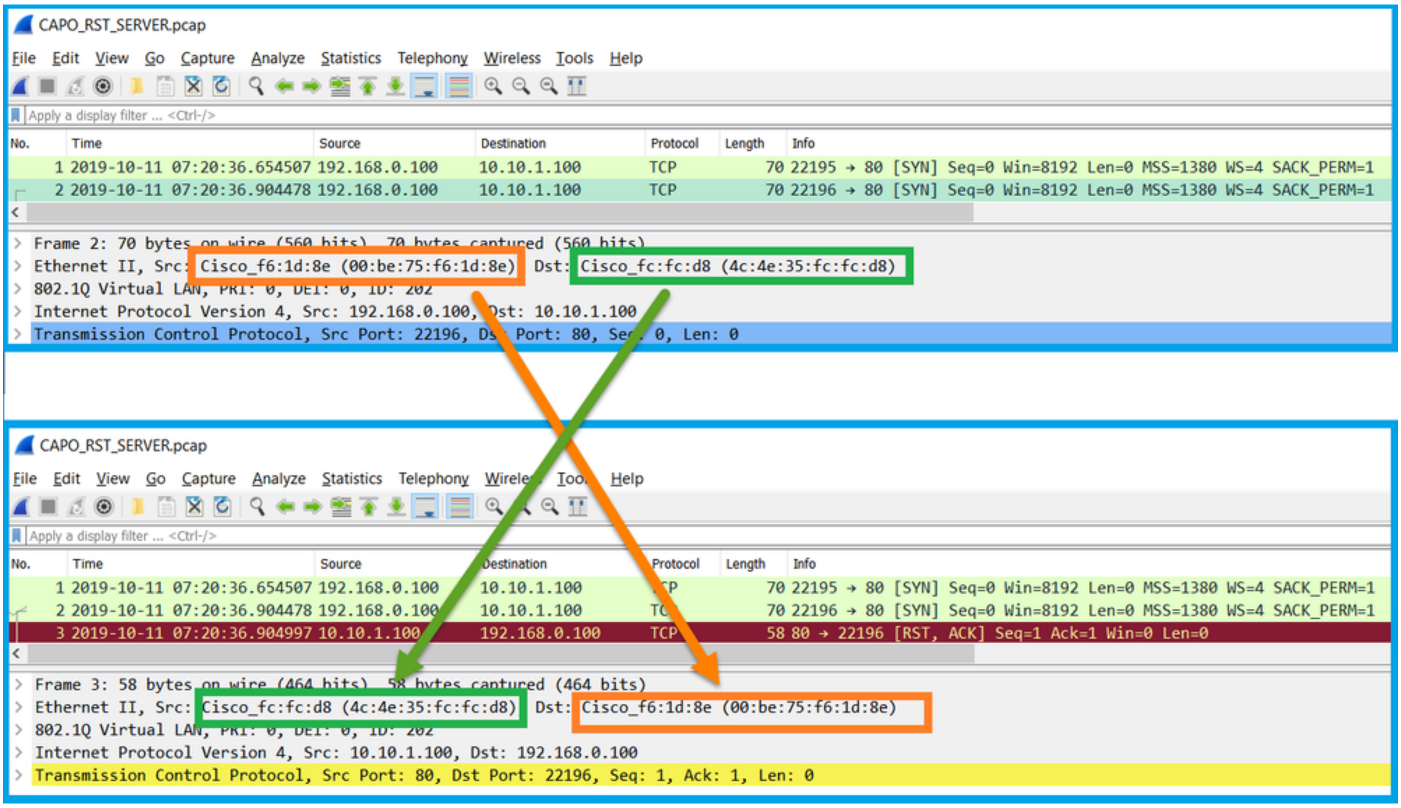
- 防火牆與適當的上游和下游裝置「通訊」（基於MAC地址）

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.檢查傳送TCP RST的源MAC地址。

確認TCP SYN封包中看到的目的地MAC與TCP RST封包中顯示的來源MAC相同。


CAPO_RST_SERVER.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time                        Source          Destination   Protocol Length Info
   1 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
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

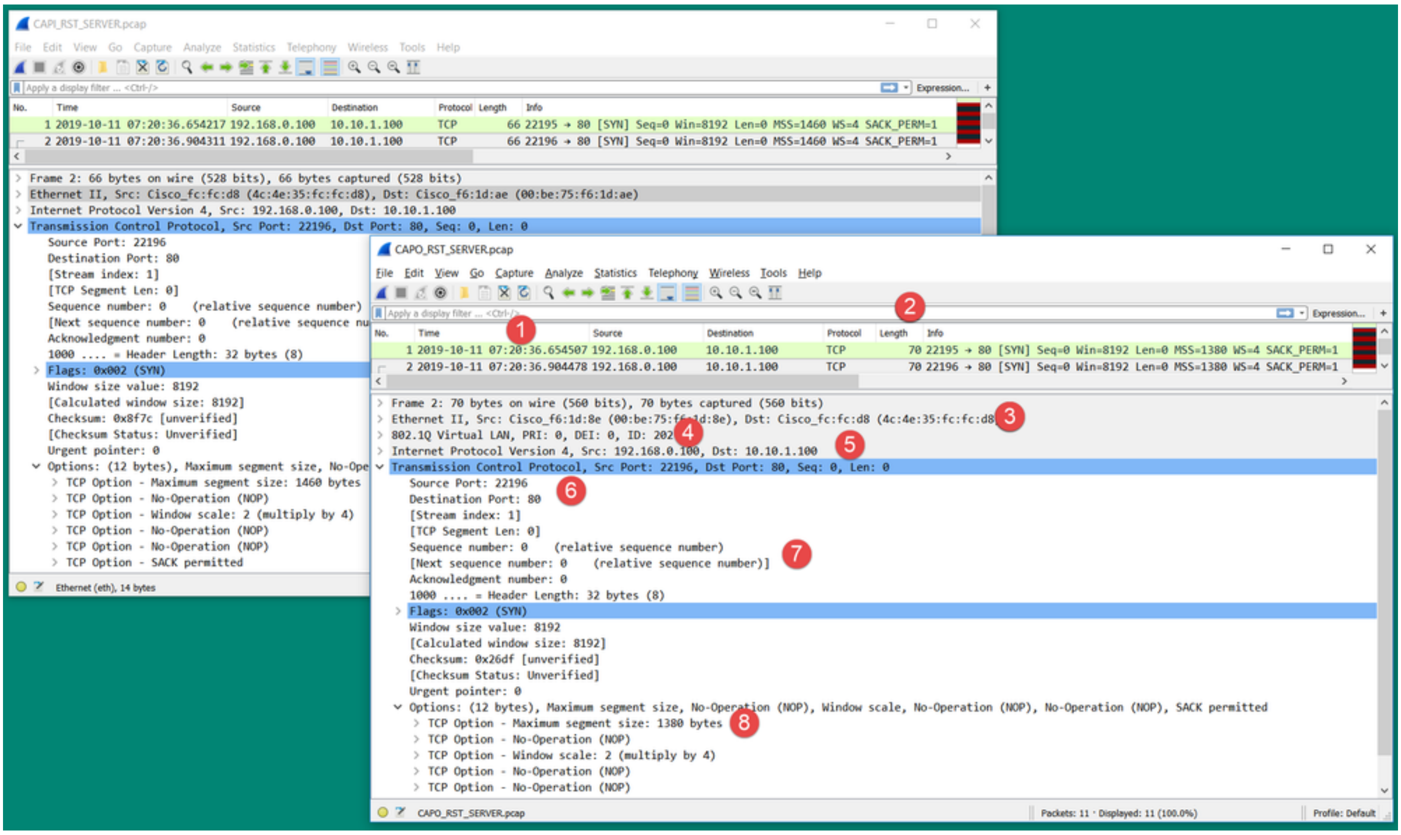
```
No.  Time                        Source          Destination     Protocol Length Info
   1 2019-10-11 07:20:36.654507 192.168.0.100   10.10.1.100     TCP      70 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
   2 2019-10-11 07:20:36.904478 192.168.0.100   10.10.1.100     TCP      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
```

此檢查旨在確認兩件事：

- 驗證沒有非對稱流。
- 檢驗MAC是否屬於預期的上游裝置。

行動2.比較入口和出口資料包。

目測比較Wireshark上的兩個資料包，驗證防火牆沒有修改/損壞這些資料包。一些預期差異被突出顯示。


CAPI_RST_SERVER.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
CAPO_RST_SERVER.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
1
2
3
4
5
6
7
8

```
No.  Time                       Source          Destination   Protocol Length Info
   1 2019-10-11 07:20:36.654217 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 2019-10-11 07:20:36.904311 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

> Frame 2: 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
v Transmission Control Protocol, Src Port: 22196, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 22196
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence nu
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0x8f7c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  v Options: (12 bytes), Maximum segment size, No-Ope
      > TCP Option - Maximum segment size: 1460 bytes
      > TCP Option - No-Operation (NOP)
      > TCP Option - Window scale: 2 (multiply by 4)
      > TCP Option - No-Operation (NOP)
      > TCP Option - No-Operation (NOP)
      > TCP Option - SACK permitted

Ethernet (eth), 14 bytes
```

```
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
v Transmission Control Protocol, Src Port: 22196, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 22196
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0x26df [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  v 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: 2 (multiply by 4)
      > TCP Option - No-Operation (NOP)
      > TCP Option - No-Operation (NOP)

CAPO_RST_SERVER.pcap          Packets: 11 · Displayed: 11 (100.0%)          Profile: Default
```

重點：

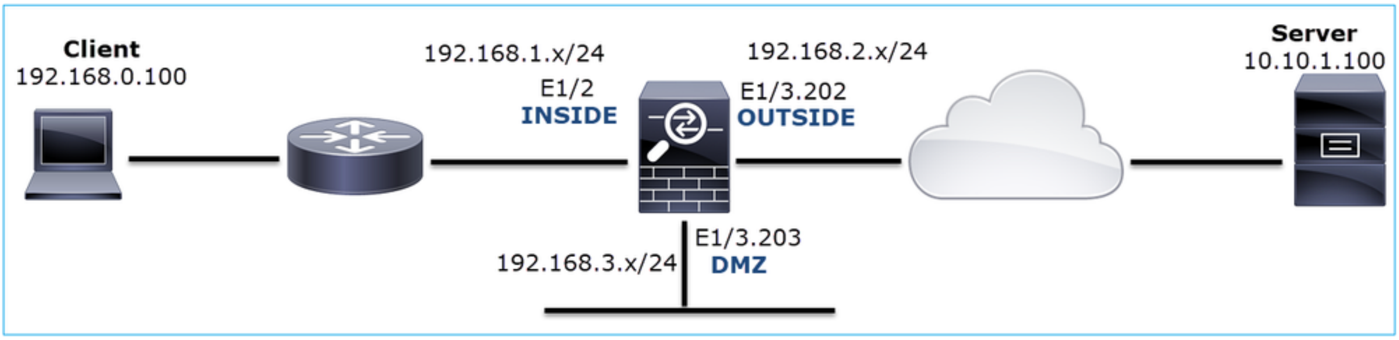
1. 時間戳不同。另一方面，這種差異必須小而合理。這取決於應用於資料包的功能和策略檢查以及裝置上的負載。
2. 資料包的長度可能會有所不同，尤其是如果防火牆僅在一端新增/刪除了dot1Q報頭。
3. MAC地址不同。
4. 如果捕獲是在子介面上進行的，則可以使用dot1Q報頭。
5. 在將NAT或埠地址轉換(PAT)應用於資料包時，IP地址是不同的。
6. 如果將NAT或PAT應用於資料包，則源埠或目標埠不同。
7. 如果禁用Wireshark Relative Sequence Number選項，就會看到由於初始序列號(ISN)隨機化，防火牆修改了TCP序列號/確認號。
8. 某些TCP選項可能被覆蓋。例如，防火牆預設會將TCP最大區段大小(MSS)變更為1380，以避免傳輸路徑中的封包分段。

行動3.在目標處執行捕獲。

如果可能，在目的地本身進行捕獲。如果無法實現，則使捕獲儘可能靠近目標。這裡的目標是驗證誰傳送了TCP RST（是目的地伺服器還是路徑中的其他裝置？）。

案例3.來自一個終端的TCP三次握手+ RST

下圖顯示拓撲：



問題說明：HTTP無法正常工作

受影響的流：

源IP:192.168.0.100

Dst IP:10.10.1.100

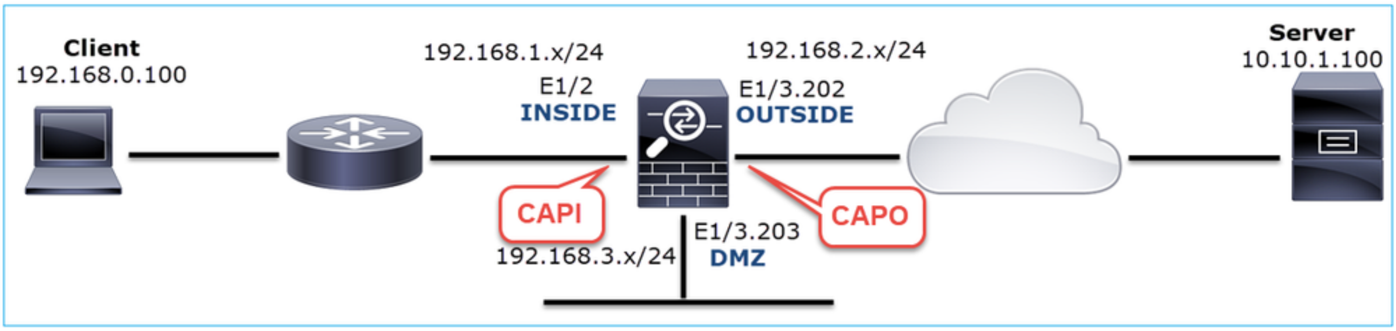
協定：TCP 80

捕獲分析

在FTD LINA引擎上啟用擷取。

<#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
```

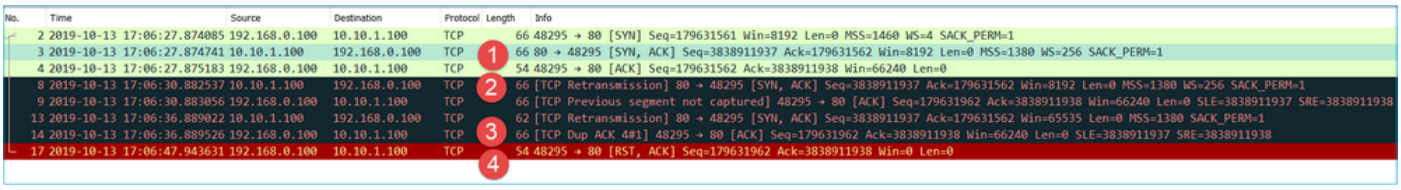


捕獲 — 非功能場景：

此問題可通過幾種不同的方式在捕獲中表現出來。

3.1 — 客戶端的TCP三次握手+延遲RST

防火牆會擷取CAPI和CAPO包含相同的封包，如下圖所示。



```
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
```

重點：

1. TCP三次握手會通過防火牆。
2. 伺服器重新傳輸SYN/ACK。
3. 客戶端重新傳輸ACK。
4. 大約20秒後，客戶端放棄並傳送TCP RST。

建議的操作

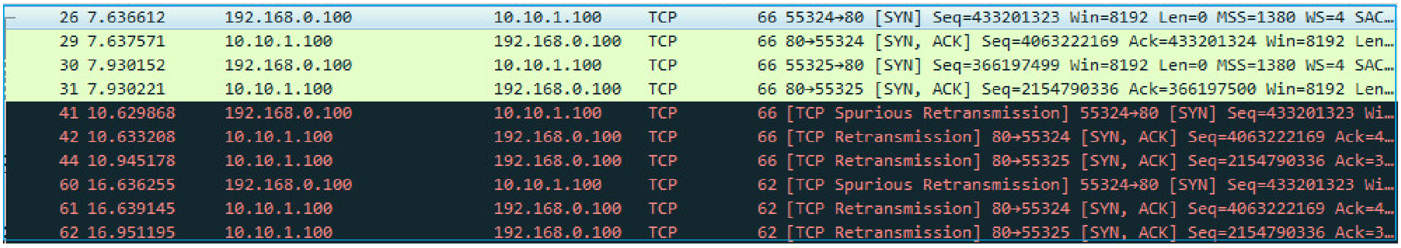
本節所列的行動旨在進一步縮小問題範圍。

操作1.儘可能靠近兩個端點捕獲捕獲。

防火牆捕獲指示伺服器未處理客戶端ACK。這是基於以下事實：

- 伺服器重新傳輸SYN/ACK。
- 客戶端重新傳輸ACK。
- 客戶端在任何資料之前傳送TCP RST或FIN/ACK。

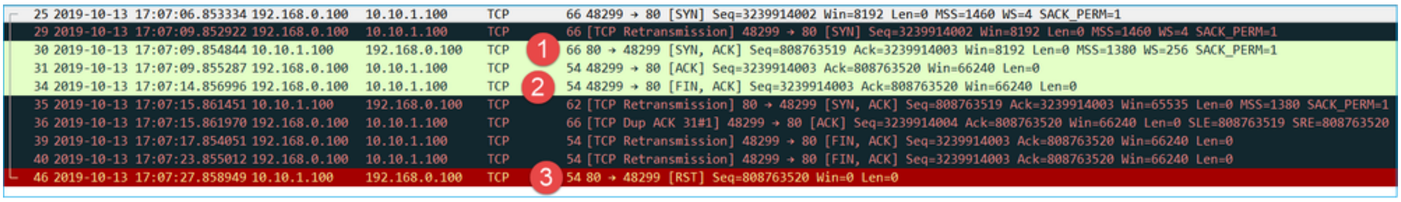
在伺服器上捕獲會顯示問題。TCP三次握手的客戶端ACK從未到達：



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

3.2 - TCP三次握手+來自客戶端的延遲FIN/ACK +來自伺服器的延遲RST

防火牆會擷取CAPI和CAPO包含相同的封包，如下圖所示。



```
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
```

重點：

1. TCP三次握手會通過防火牆。
2. 約5秒後，客戶端傳送FIN/ACK。
3. 大約20秒後，伺服器放棄並傳送TCP RST。

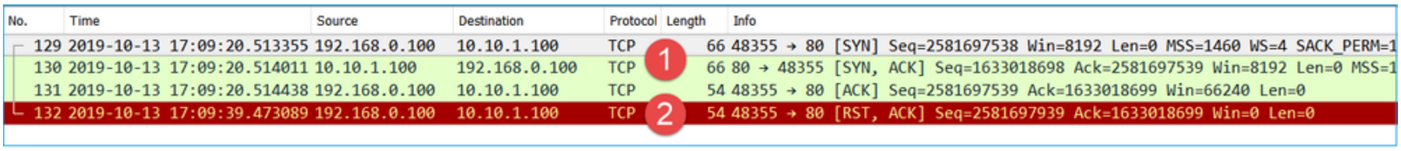
根據捕獲結果，可以推斷出，雖然存在通過防火牆的TCP三次握手，但似乎在一個端點上從未真正完成握手（重新傳輸表示此情況）。

建議的操作

與案例3.1相同

3.3 — 客戶端的TCP三次握手+延遲RST

防火牆會擷取CAPI和CAPO包含相同的封包，如下圖所示。



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

重點：

1. TCP三次握手會通過防火牆。
2. 大約20秒後，客戶端放棄並傳送TCP RST。

根據這些捕獲可以得出結論：

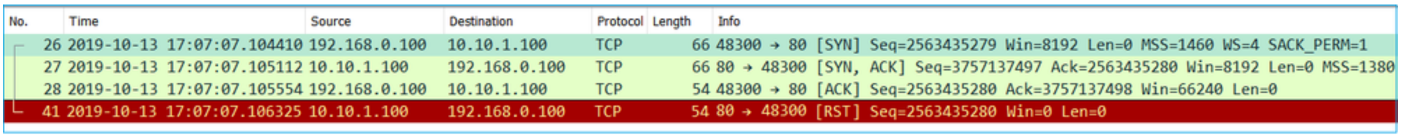
- 5-20秒後，一個終端放棄並決定終止連線。

建議的操作

與案例3.1相同

3.4 — 來自伺服器的TCP三次握手+即時RST

兩個防火牆都會擷取CAPI，而CAPI包含這些封包，如下圖所示。



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

重點：

1. TCP三次握手會通過防火牆。
2. 在ACK封包過後幾毫秒時，伺服器會產生TCP RST。

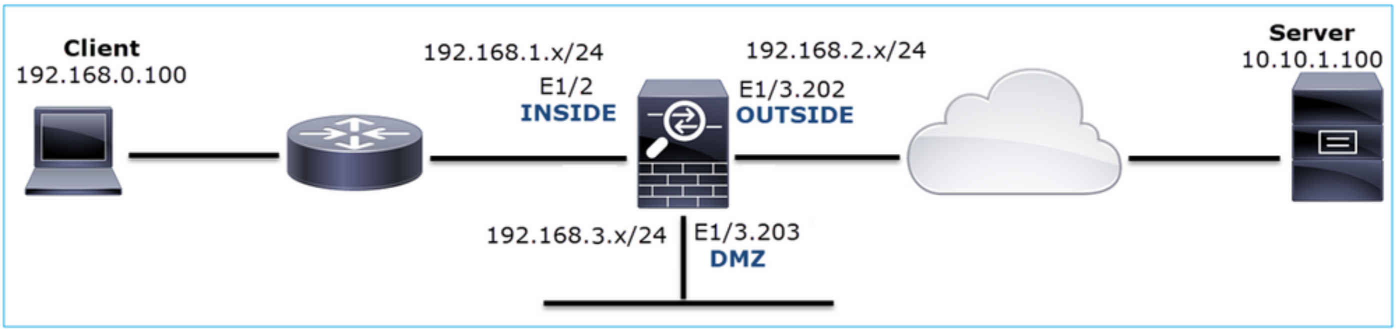
建議的操作

操作：儘可能在靠近伺服器的位置捕獲捕獲。

來自伺服器的立即TCP RST可能表示伺服器故障或傳送TCP RST的路徑中的裝置。在伺服器本身進行捕獲，確定TCP RST的來源。

案例4.來自使用者端的TCP RST

下圖顯示拓撲：



問題說明：HTTP無法正常工作。

受影響的流：

源IP:192.168.0.100

Dst IP:10.10.1.100

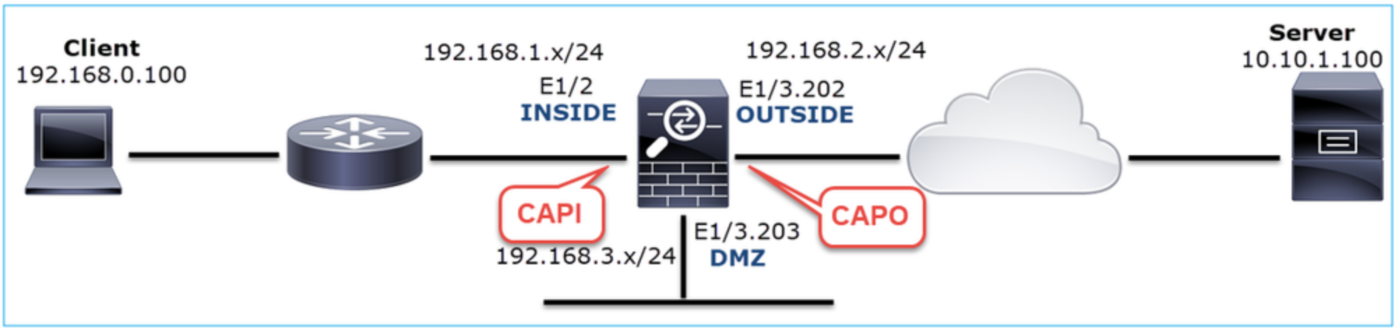
協定：TCP 80

捕獲分析

在FTD LINA引擎上啟用擷取。

<#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
```



捕獲 — 非功能場景：

以下是CAPI的內容。

<#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
```

以下是CAPO內容：

<#root>

```
firepower#

show capture CAPO


11 packets captured

   1: 12:32:22.860780   802.1Q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: S 1386249852:138624985
   2: 12:32:23.111429   802.1Q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: S 3000518857:300051885
   3: 12:32:23.112405   802.1Q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: R 3514091874:351409187
   4: 12:32:25.858125   802.1Q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: S 1386249852:138624985
   5: 12:32:25.868729   802.1Q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: R 2968892337:296889233
   6: 12:32:26.108240   802.1Q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: S 3822259745:382225974
   7: 12:32:26.109094   802.1Q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: R 40865466:40865466(0)
   8: 12:32:31.860062   802.1Q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: S 4294058752:429405875
   9: 12:32:31.860917   802.1Q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: R 1581733941:158173394
  10: 12:32:32.160102   802.1Q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: S 4284301197:428430119
  11: 12:32:32.160971   802.1Q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: R 502906918:502906918(
11 packets shown
```

防火牆日誌顯示：

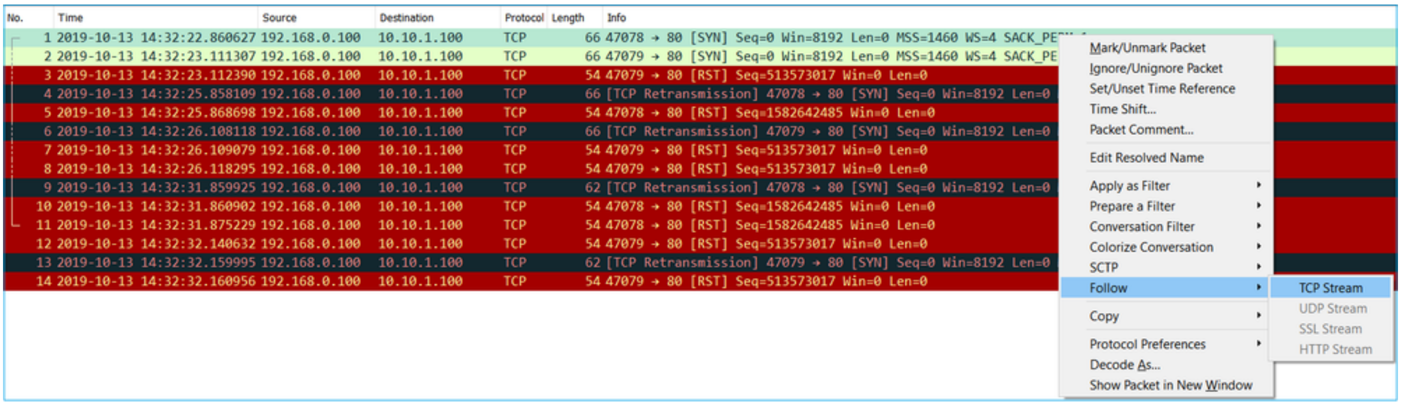
<#root>

```
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
```

這些日誌指示存在到達防火牆INSIDE介面的TCP RST

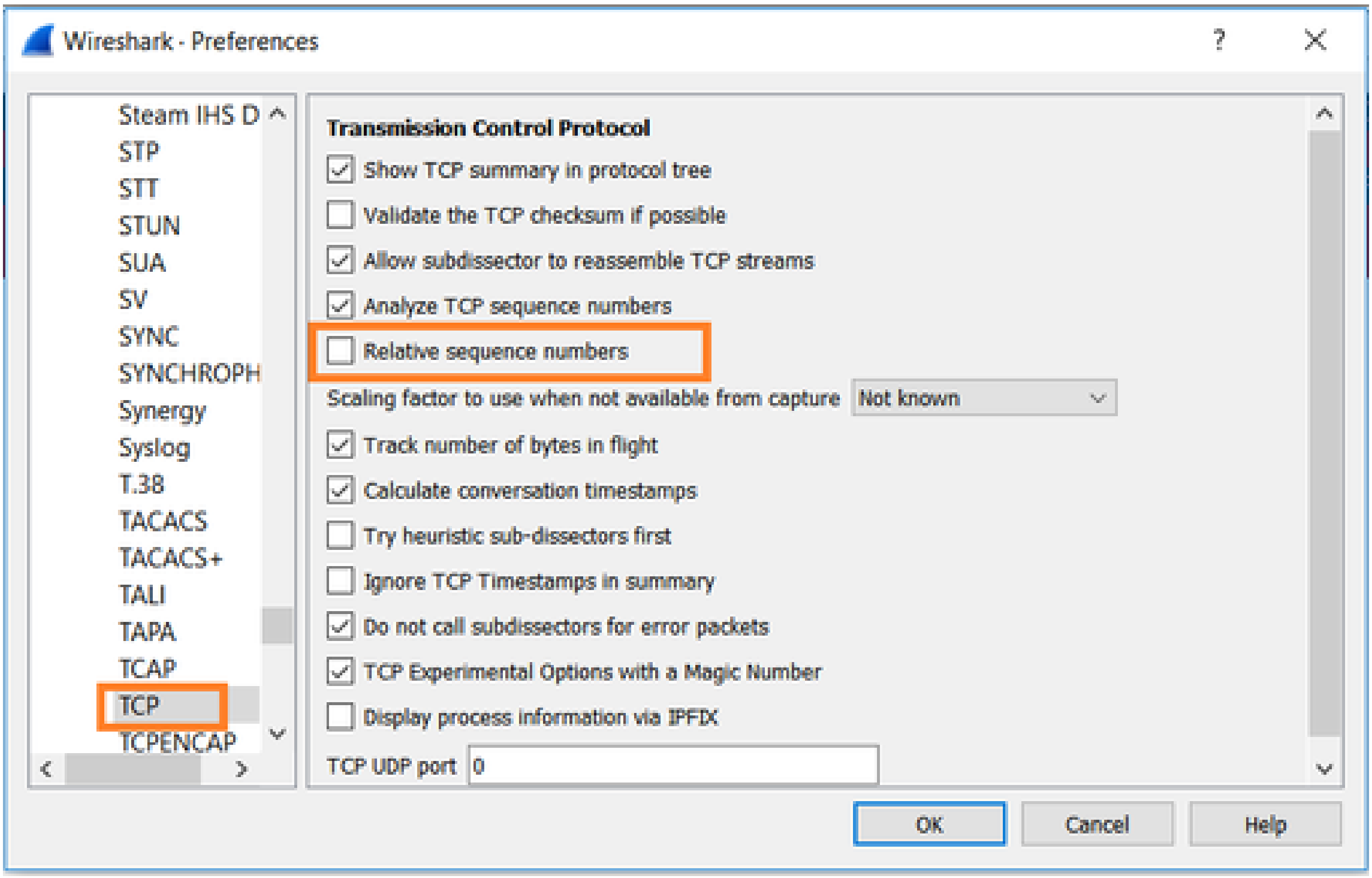
Wireshark中的CAPI捕獲：

如圖所示，跟隨第一個TCP資料流。



```
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 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 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 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 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 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 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 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 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 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 47079 → 80 [RST] Seq=513573017 Win=0 Len=0
```

在Wireshark下，導航到編輯>首選項>協定> TCP，然後取消選擇Relative sequence numbers選項，如下圖所示。

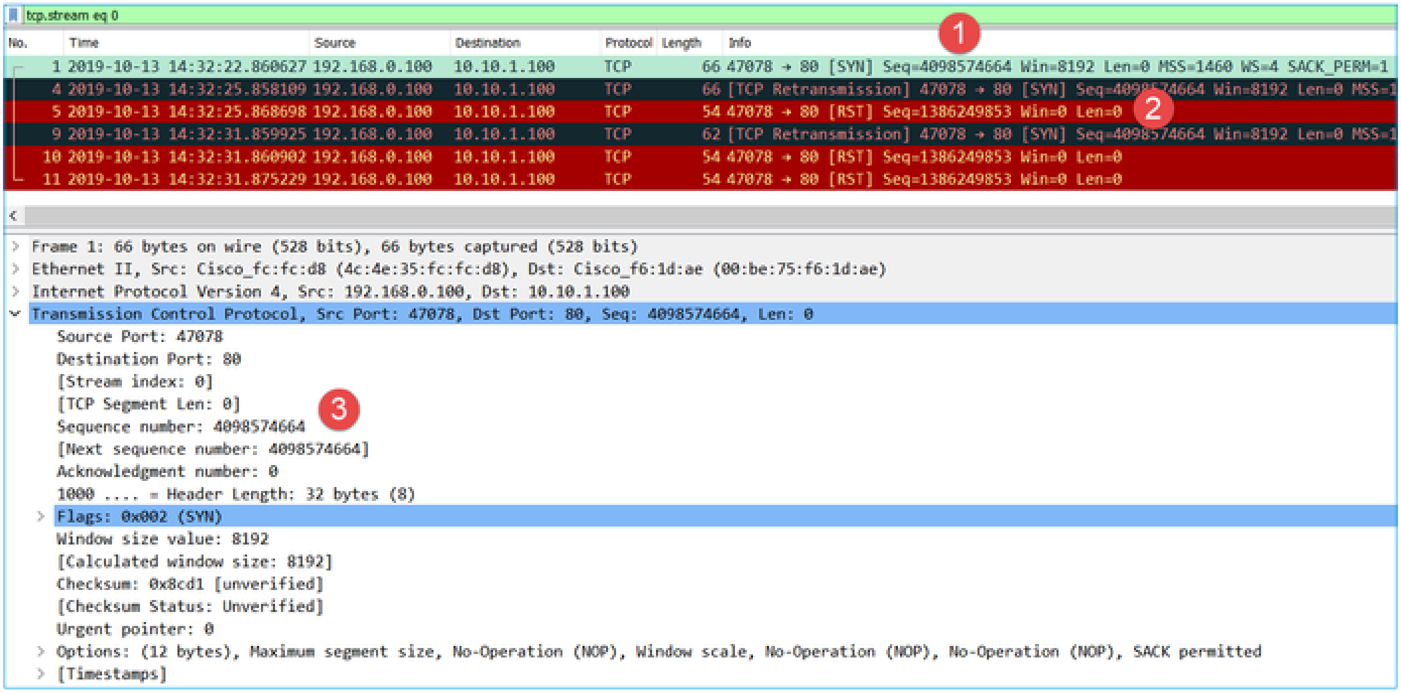


Wireshark - Preferences

Transmission Control Protocol

☑ Show TCP summary in protocol tree
□ Validate the TCP checksum if possible
☑ Allow subdissector to reassemble TCP streams
☑ Analyze TCP sequence numbers
□ Relative sequence numbers
Scaling factor to use when not available from capture: Not known
☑ Track number of bytes in flight
☑ Calculate conversation timestamps
□ Try heuristic sub-dissectors first
□ Ignore TCP Timestamps in summary
☑ Do not call subdissectors for error packets
☑ TCP Experimental Options with a Magic Number
□ Display process information via IPFIX
TCP UDP port: 0

此圖顯示CAPI擷取中第一個流程的內容：



tcp.stream eq 0

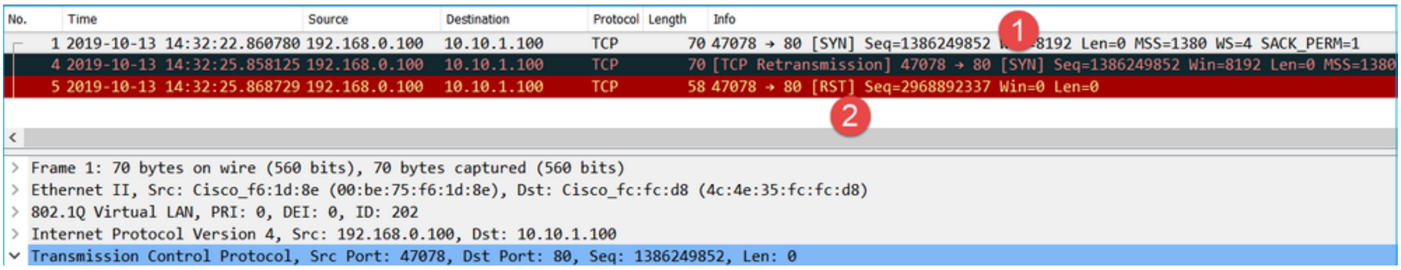
```
No.  Time                        Source         Destination  Protocol Length Info
   1 2019-10-13 14:32:22.860627 192.168.0.100  10.10.1.100  TCP      66 47078 → 80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
   4 2019-10-13 14:32:25.858109 192.168.0.100  10.10.1.100  TCP      66 [TCP Retransmission] 47078 → 80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1
   5 2019-10-13 14:32:25.868698 192.168.0.100  10.10.1.100  TCP      54 47078 → 80 [RST] Seq=1386249853 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=4098574664 Win=8192 Len=0 MSS=1
  10 2019-10-13 14:32:31.860902 192.168.0.100  10.10.1.100  TCP      54 47078 → 80 [RST] Seq=1386249853 Win=0 Len=0
  11 2019-10-13 14:32:31.875229 192.168.0.100  10.10.1.100  TCP      54 47078 → 80 [RST] Seq=1386249853 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
v Transmission Control Protocol, Src Port: 47078, Dst Port: 80, Seq: 4098574664, Len: 0
    Source Port: 47078
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 4098574664
    [Next sequence number: 4098574664]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0x8cd1 [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
  > [Timestamps]
```

重點：

1. 使用者端傳送TCP SYN封包。
2. 使用者端傳送TCP RST封包。
3. TCP SYN資料包的序列號值等於4098574664。

CAPO捕獲中的同一流包含：



```
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 W  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_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
v Transmission Control Protocol, Src Port: 47078, Dst Port: 80, Seq: 1386249852, Len: 0
```

重點：

1. 使用者端傳送TCP SYN封包。防火牆隨機化ISN。
2. 使用者端傳送TCP RST封包。

根據兩個擷取可得出以下結論：

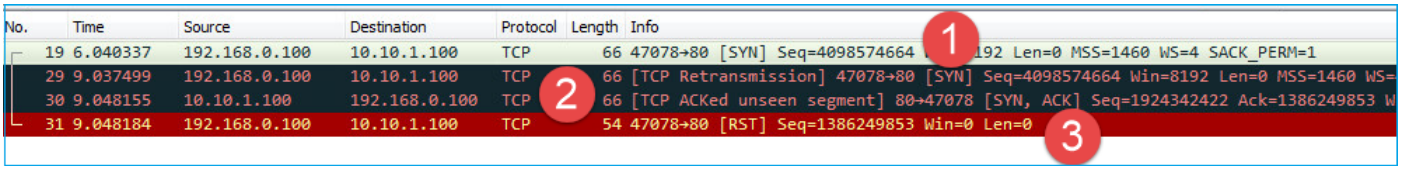
- 客戶端和伺服器之間沒有TCP三次握手。
- 有一個來自使用者端的TCP RST。CAPI捕獲中的TCP RST序列號值為1386249853。

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.在客戶端上執行捕獲。

根據在防火牆上收集的擷取，有強烈的跡象顯示非對稱流量。這是基於使用者端傳送值為1386249853的TCP RST（隨機化ISN）這一事實：

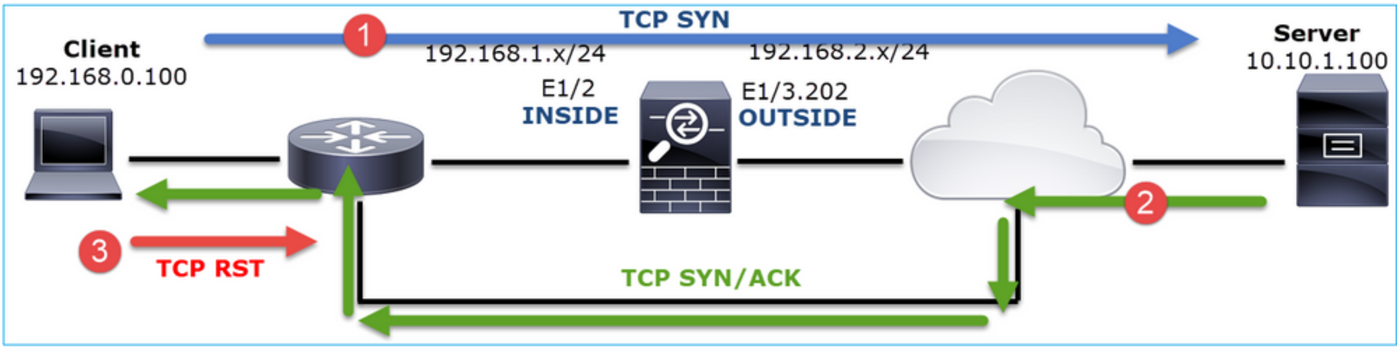


```
No.  Time      Source         Destination    Protocol Length Info
 19 6.040337  192.168.0.100  10.10.1.100    TCP      66 47078→80 [SYN] Seq=4098574664    192 Len=0 MSS=1460 WS=4 SACK_PERM=1
 29 9.037499  192.168.0.100  10.10.1.100    TCP      66 [TCP Retransmission] 47078→80 [SYN] Seq=4098574664 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
```

重點：

1. 使用者端傳送TCP SYN封包。序列號為4098574664，與防火牆INSIDE介面(CAPI)上顯示的序列號相同
2. 有一個ACK編號為1386249853的TCP SYN/ACK（預計因為ISN隨機化）。在防火牆擷取中看不到此封包
3. 使用者端傳送一個TCP RST，因為它預期的SYN/ACK的ACK編號值為4098574665，但收到的值為1386249853

其視覺化結果為：

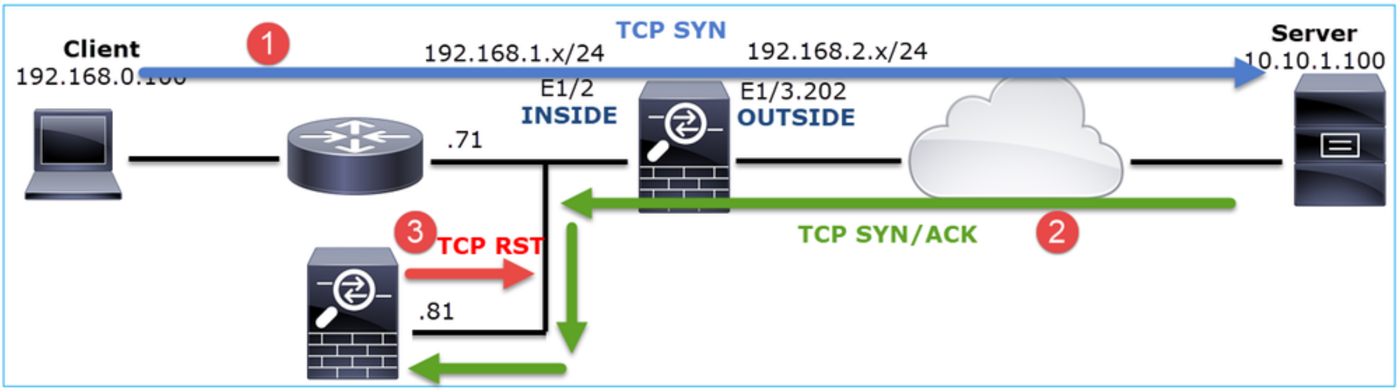


行動2.檢查客戶端和防火牆之間的路由。

確認：

- 捕獲中看到的MAC地址是預期的MAC地址。
- 確保防火牆和客戶端之間的路由是對稱的。

某些情況下，RST來自位於防火牆和客戶端之間的裝置，而內部網路中存在非對稱路由。以下為典型案例：



在這種情況下，捕獲包含此內容。請注意TCP SYN封包的來源MAC位址與TCP RST的來源MAC位址以及TCP SYN/ACK封包的目的地MAC位址之間的差異：

<#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)
...
```

案例5.TCP傳輸緩慢（場景1）

問題描述：

主機10.11.4.171和10.77.19.11之間的SFTP傳輸很慢。雖然兩台主機之間的最小頻寬(BW)為100 Mbps，但傳輸速度不會超過5 Mbps。

同時，主機10.11.2.124和172.25.18.134之間的傳輸速度相當高。

背景理論：

單個TCP流的最大傳輸速度由頻寬延遲產品(BDP)決定。使用的公式如下圖所示：

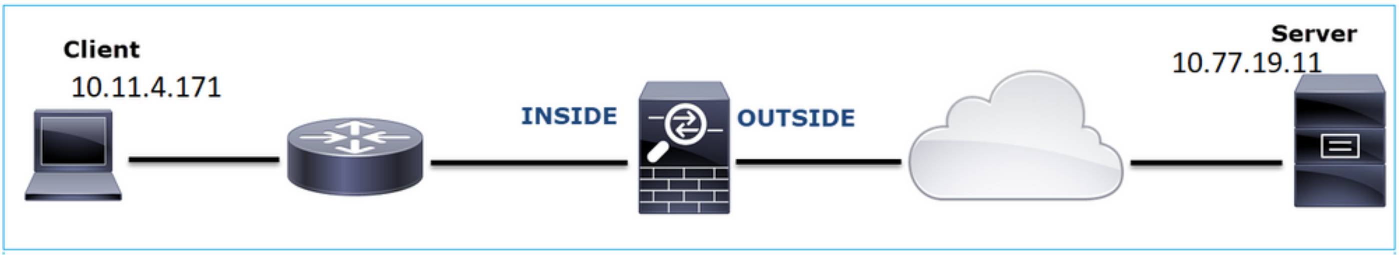
$$\text{Max Single TCP Flow Throughput [bps]} = \frac{\text{TCP Window (Bytes)}}{\text{RTT (Seconds)}} \text{ x 8 [bits/Byte]}$$

有關BDP的更多詳細資訊，請在此處檢視資源：

- 為什麼即使鏈路為1Gbps，您的應用程式也只使用10Mbps?
- BRKSEC-3021 — 高級 — 最大化防火牆效能

案例 1.傳輸緩慢

下圖顯示拓撲：



受影響的流：

源IP:10.11.4.171

Dst IP:10.77.19.11

協定：SFTP（使用SSH的FTP）

捕獲分析

在FTD LINA引擎上啟用擷取：

<#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和FP21xx捕獲上的LINA捕獲會影響通過FTD的流量的傳輸速率。排解效能（透過FTD的傳輸緩慢）疑難問題時，請勿在FP1xxx和FP21xxx平台上啟用LINA擷取。除了在來源和目的地主機上進行擷取外，還應使用SPAN或硬體分流器裝置。此問題已記錄在Cisco錯誤ID CSCvo30697中

.

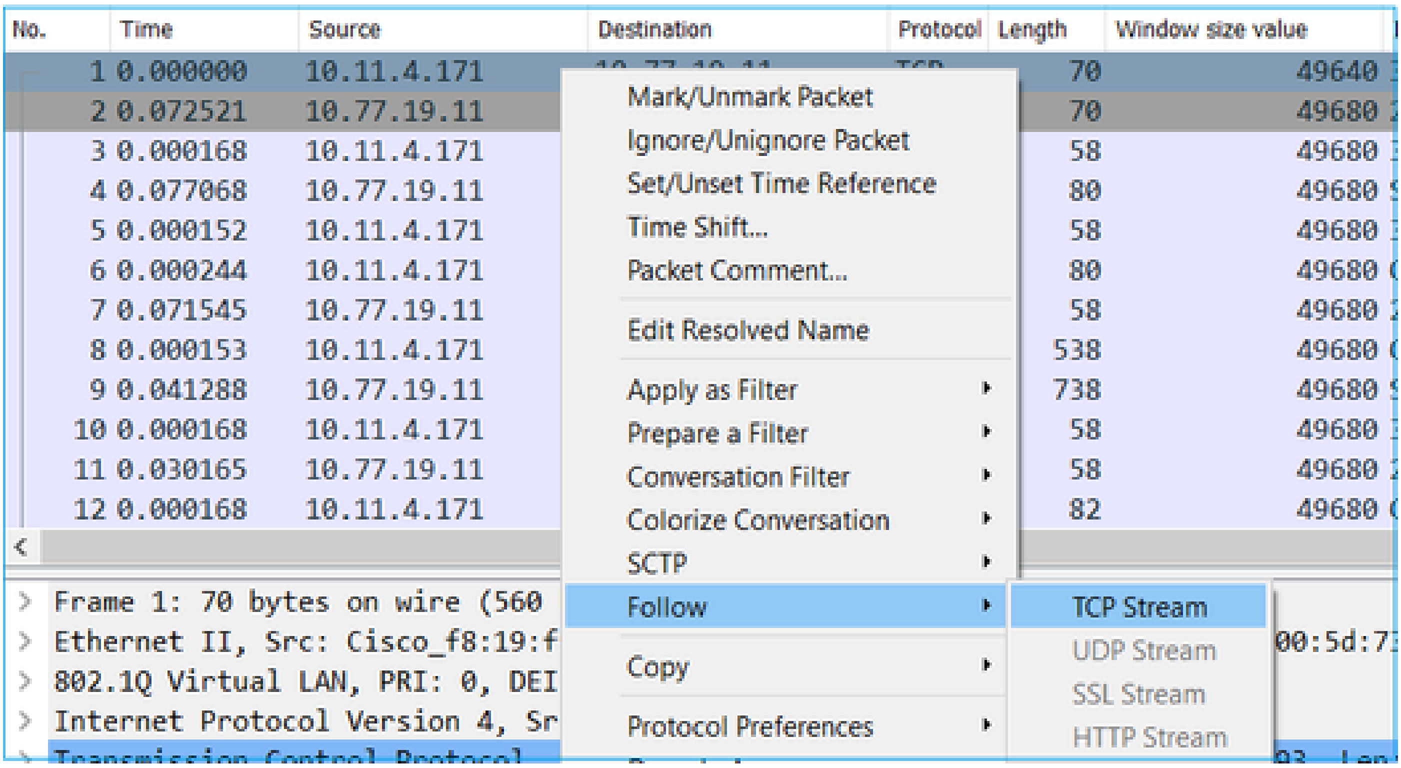
```
<#root>
firepower#
capture CAPI type raw-data trace interface inside match icmp any any
WARNING: Running packet capture can have an adverse impact on performance.
```

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

往返時間(RTT)計算

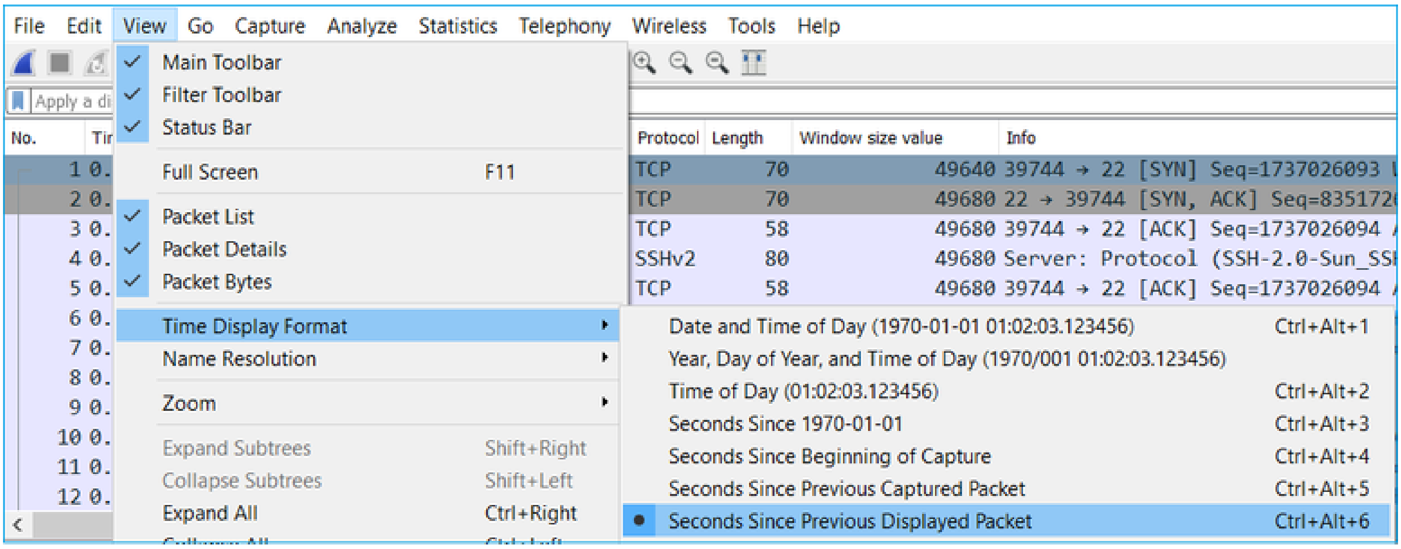
首先，確定傳輸流程並遵循該流程：



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

```
> Frame 1: 70 bytes on wire (560
> Ethernet II, Src: Cisco_f8:19:f                                   00:5d:7
> 802.1Q Virtual LAN, PRI: 0, DEI
> Internet Protocol Version 4, Sr
> Transmission Control Protocol
```

更改Wireshark檢視以顯示自上次顯示資料包以來的秒數。這樣可簡化RTT的計算：

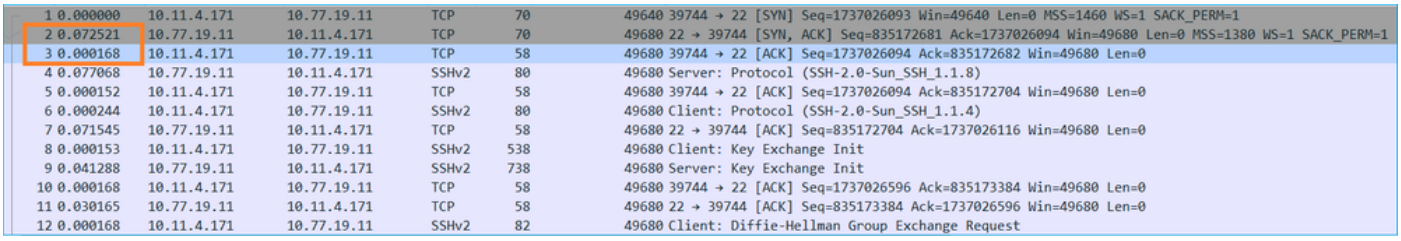


No.	Protocol	Length	Window size value	Info
1	TCP	70	49640	39744 → 22 [SYN] Seq=1737026093
2	TCP	70	49680	22 → 39744 [SYN, ACK] Seq=835172
3	TCP	58	49680	39744 → 22 [ACK] Seq=1737026094
4	SSHv2	80	49680	Server: Protocol (SSH-2.0-Sun_SS
5	TCP	58	49680	39744 → 22 [ACK] Seq=1737026094

● Seconds Since Previous Displayed Packet Ctrl+Alt+6

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a di
✓ Main Toolbar
✓ Filter Toolbar
✓ Status Bar
Full Screen F11
✓ Packet List
✓ Packet Details
✓ Packet Bytes
Time Display Format
Name Resolution
Zoom
Expand Subtrees Shift+Right
Collapse Subtrees Shift+Left
Expand All Ctrl+Right
Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+1
Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
Time of Day (01:02:03.123456) Ctrl+Alt+2
Seconds Since 1970-01-01 Ctrl+Alt+3
Seconds Since Beginning of Capture Ctrl+Alt+4
Seconds Since Previous Captured Packet Ctrl+Alt+5

RTT可通過在2個封包交換（一個朝向來源，一個朝向目的地）之間加上時間值來計算。在這種情況下，封#2連線會顯示防火牆與傳送SYN/ACK封包的裝置（伺服器）之間的RTT。Packet #3顯示防火牆與傳送ACK封包的裝置（使用者端）之間的RTT。將兩個數字相加可很好地估計端到端RTT:

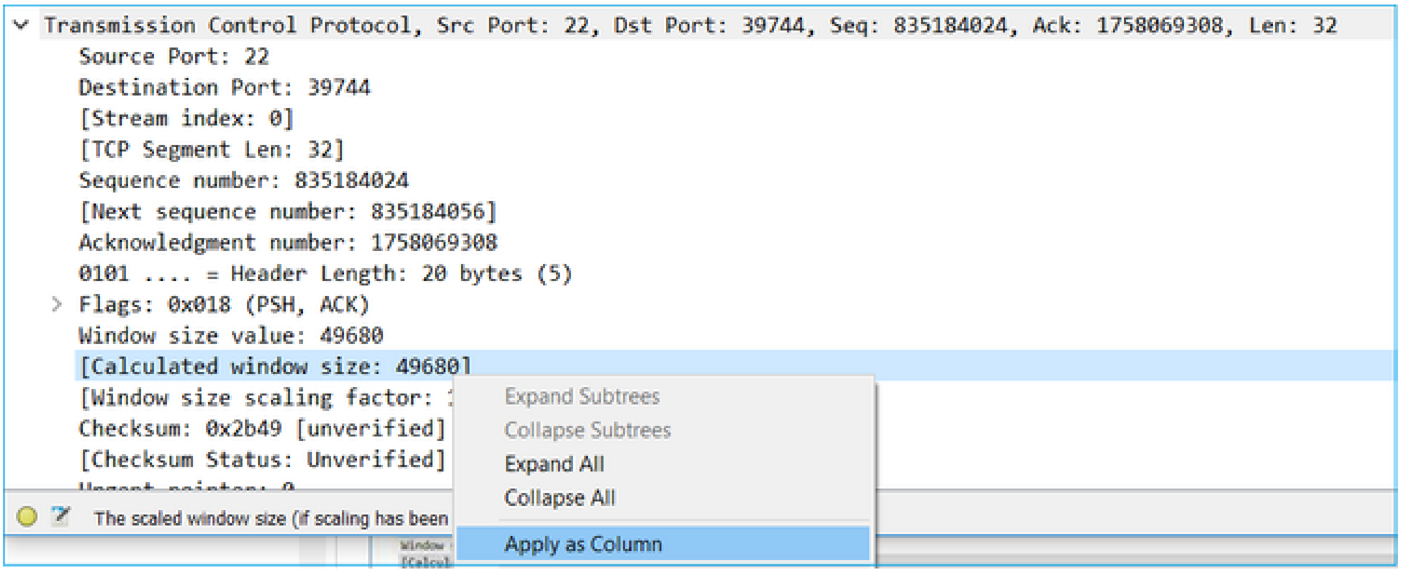


```
 1 0.000000  10.11.4.171   10.77.19.11   TCP     70   49640 39744 → 22 [SYN] Seq=1737026093 Win=49640 Len=0 MSS=1460 WS=1 SACK_PERM=1
 2 0.072521  10.77.19.11   10.11.4.171   TCP     70   49680 22 → 39744 [SYN, ACK] Seq=835172681 Ack=1737026094 Win=49680 Len=0 MSS=1380 WS=1 SACK_PERM=1
 3 0.000168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

RTT ≈ 80毫秒

TCP視窗大小計算

展開TCP資料包，展開TCP報頭，選擇Calculated window size，然後選擇Apply as Column:

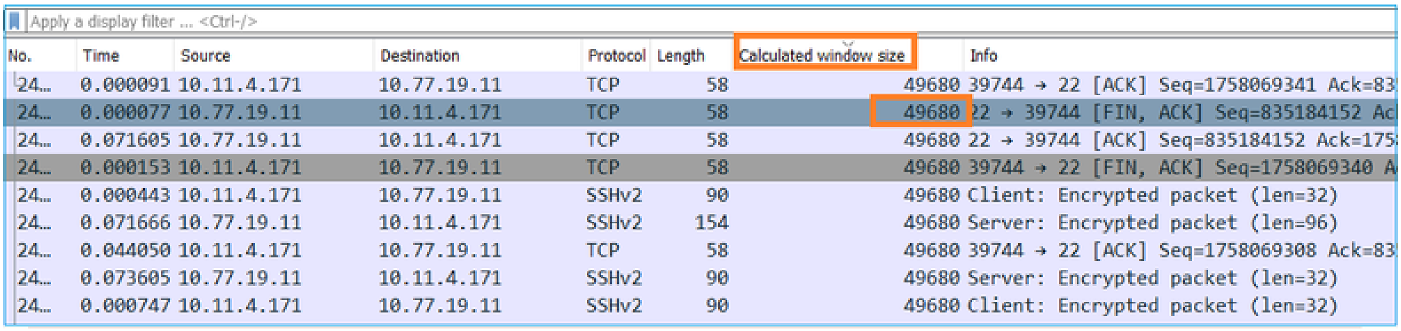


```
∨ Transmission Control Protocol, Src Port: 22, Dst Port: 39744, Seq: 835184024, Ack: 1758069308, Len: 32
    Source Port: 22
    Destination Port: 39744
    [Stream index: 0]
    [TCP Segment Len: 32]
    Sequence number: 835184024
    [Next sequence number: 835184056]
    Acknowledgment number: 1758069308
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 49680
    [Calculated window size: 49680]
    [Window size scaling factor: 1
    Checksum: 0x2b49 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
The scaled window size (if scaling has been
```

檢查Calculated window size value列，檢視TCP會話期間的最大視窗大小值。也可以選擇列名並對值排序。

如果測試檔案下載(server > client)，則必須檢查伺服器通告的值。伺服器通告的最大視窗大小值確定實現的最大傳輸速度。

在這種情況下，TCP視窗大小為≈ 50000 Bytes

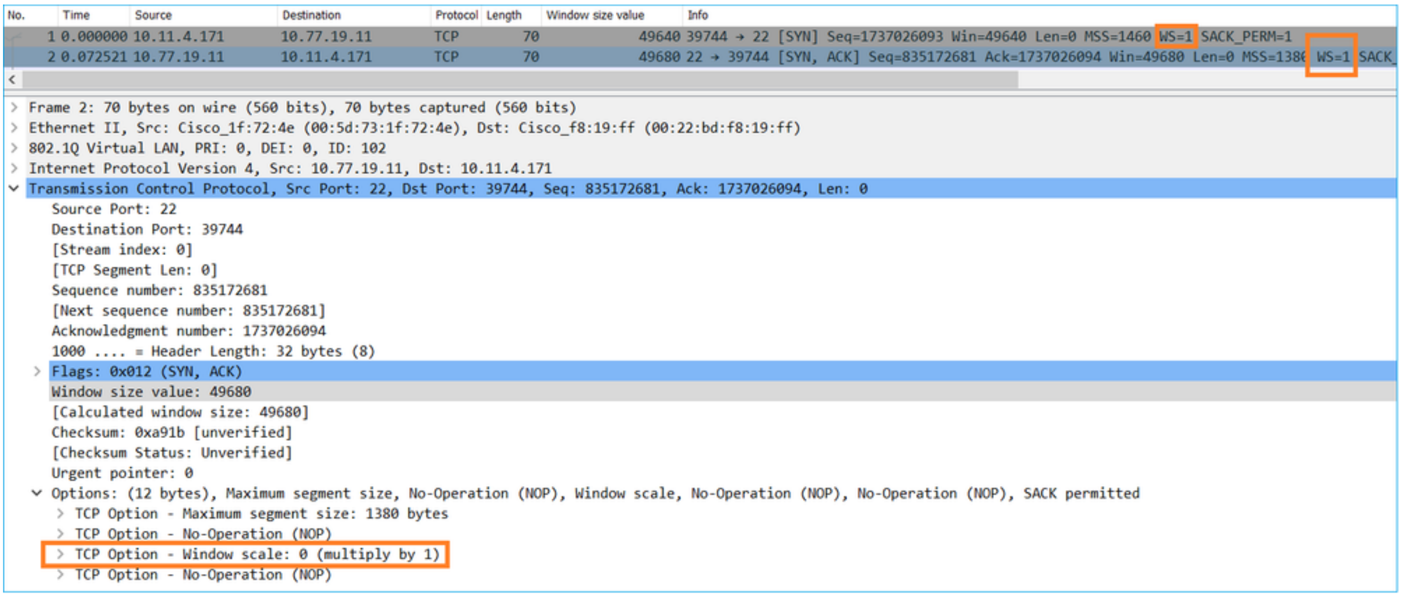


```
Apply a display filter ... <Ctrl-/>
No.   Time      Source       Destination  Protocol Length  Calculated window size  Info
24... 0.000091 10.11.4.171  10.77.19.11  TCP       58                      49680 39744 → 22 [ACK] Seq=1758069341 Ack=83
24... 0.000077 10.77.19.11  10.11.4.171  TCP       58                      49680 22 → 39744 [FIN, ACK] Seq=835184152 Ac
24... 0.071605 10.77.19.11  10.11.4.171  TCP       58                      49680 22 → 39744 [ACK] Seq=835184152 Ack=175
24... 0.000153 10.11.4.171  10.77.19.11  TCP       58                      49680 39744 → 22 [FIN, ACK] Seq=1758069340 A
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=83
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)
```

基於這些值，並使用「頻寬延遲乘積」公式，您可以獲得在這些情況下可達到的最大理論頻寬：50000*8/0.08 = 5 Mbps的最大理論頻寬。

這與客戶端在此案例中的體驗相符。

仔細檢查TCP三次握手。兩端（更重要的是伺服器）都通告視窗縮放值0，這意味著2^0 = 1（無視窗縮放）。這會對傳輸速率產生負面影響：



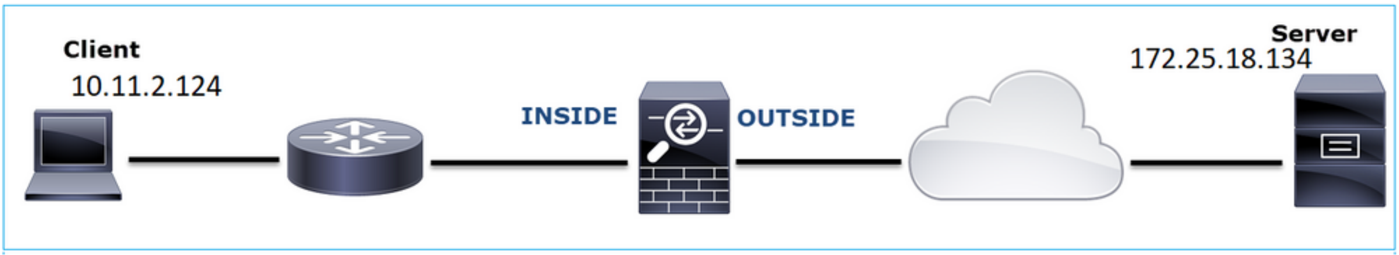
```
No.  Time     Source      Destination  Protocol Length  Window size value  Info
   1 0.000000 10.11.4.171 10.77.19.11  TCP      70      49640 39744 → 22 [SYN] Seq=1737026093 Win=49640 Len=0 MSS=1460 WS=1 SACK_PERM=1
   2 0.072521 10.77.19.11 10.11.4.171  TCP      70      49680 22 → 39744 [SYN, ACK] Seq=835172681 Ack=1737026094 Win=49680 Len=0 MSS=138( 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
v 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
  v 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)
```

此時，需要在伺服器上執行捕獲，確認是通告視窗比例= 0的捕獲者並重新配置它（有關如何執行此操作的資訊，請檢視伺服器文檔）。

案例 2.快速傳輸

現在來瞭解一下理想情況（透過同一個網路快速傳輸）：

拓撲：



利息流：

源IP:10.11.2.124

Dst IP:172.25.18.134

協定：SFTP（使用SSH的FTP）

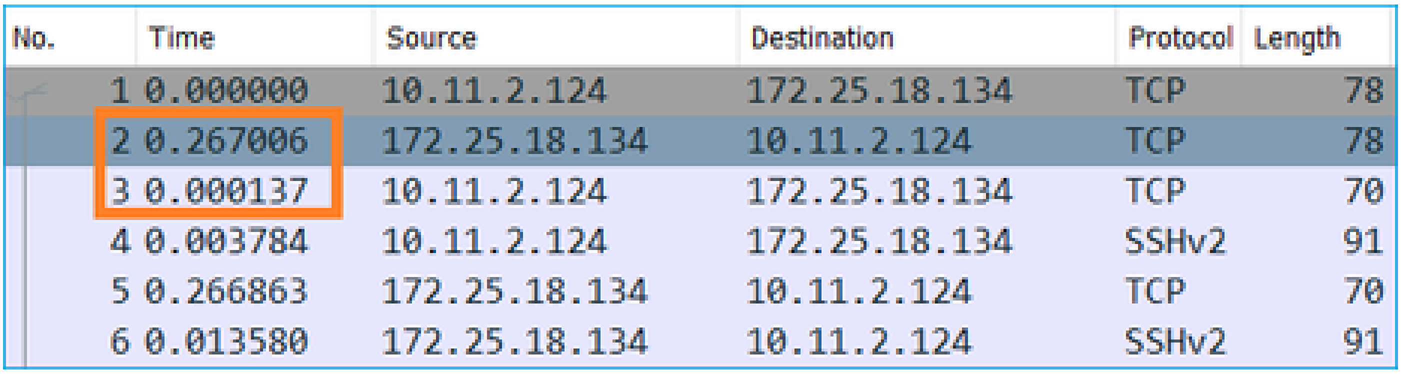
在FTD LINA引擎上啟用擷取

<#root>

```
firepower#

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

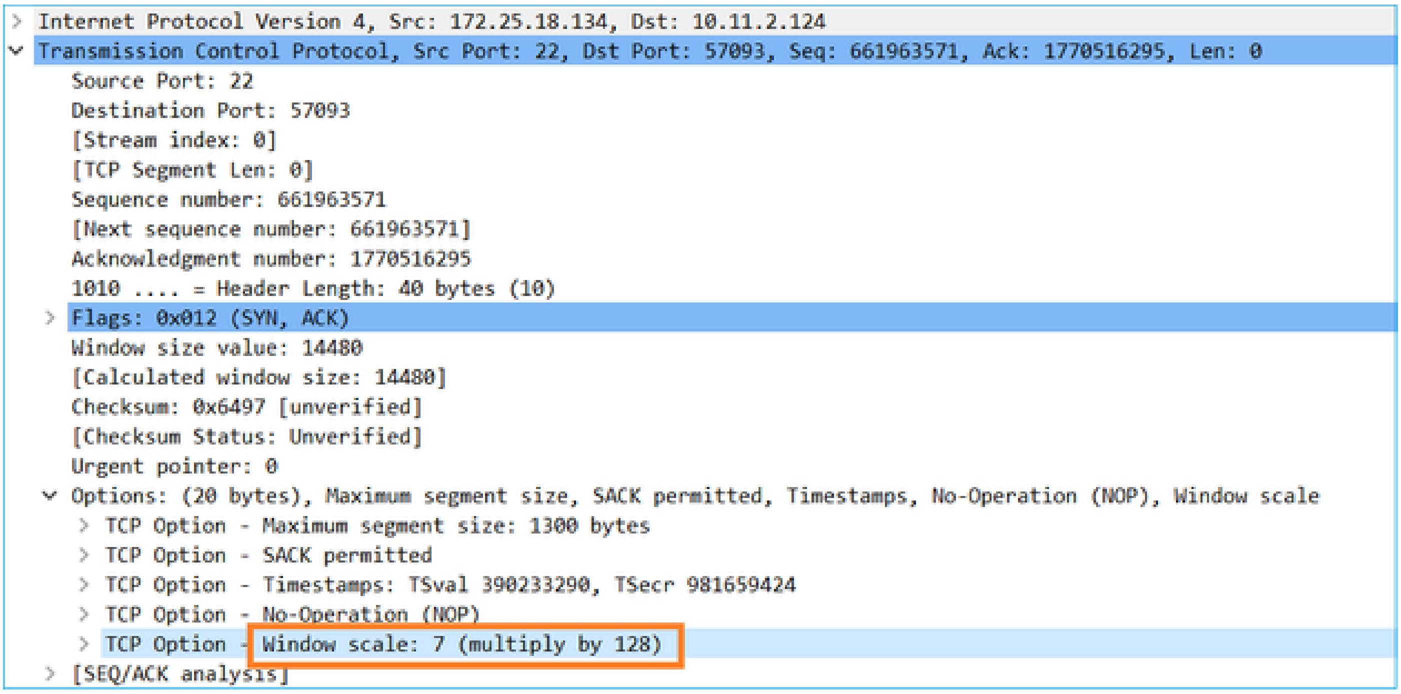
```
firepower#

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

往返時間(RTT)計算：在這種情況下，RTT為≈毫秒。



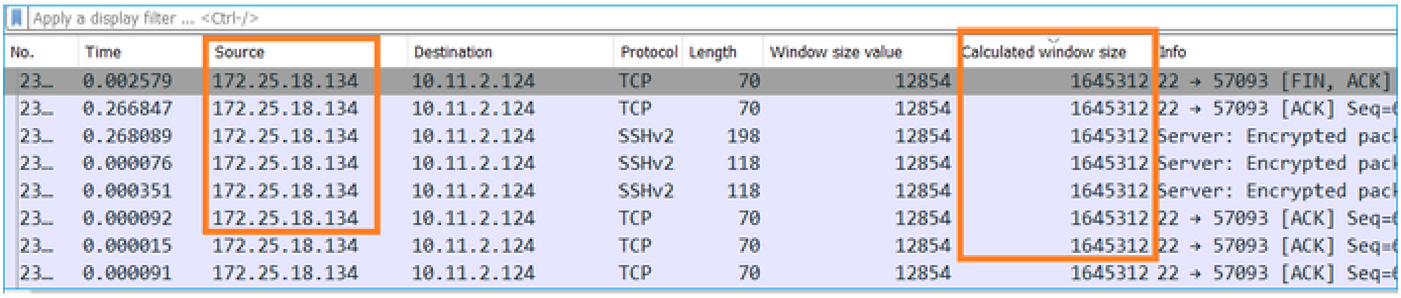
No.	Time	Source	Destination	Protocol	Length
1	0.000000	10.11.2.124	172.25.18.134	TCP	78
2	0.267006	172.25.18.134	10.11.2.124	TCP	78
3	0.000137	10.11.2.124	172.25.18.134	TCP	70
4	0.003784	10.11.2.124	172.25.18.134	SSHv2	91
5	0.266863	172.25.18.134	10.11.2.124	TCP	70
6	0.013580	172.25.18.134	10.11.2.124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

TCP視窗大小計算：伺服器通告TCP視窗比例因子7。



```
> 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]
```

伺服器的TCP視窗大小為≈ 1600000位元組：



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

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 pac
23…	0.000076	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pac
23…	0.000351	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pac
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=

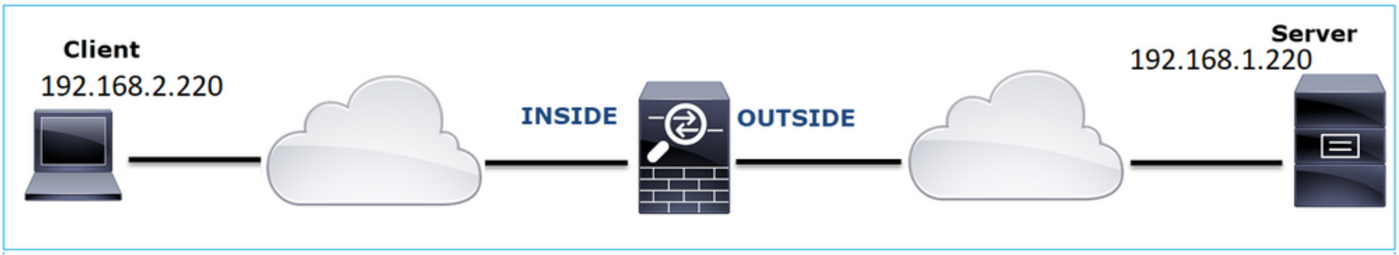
基於這些值，頻寬延遲產品公式可提供：

1600000*8/0.3 = 43 Mbps最大理論傳輸速度

案例6.TCP傳輸緩慢（案例2）

問題描述：通過防火牆的FTP檔案傳輸（下載）速度緩慢。

此圖顯示拓撲：



受影響的流：

源IP:192.168.2.220

Dst IP:192.168.1.220

協定：FTP

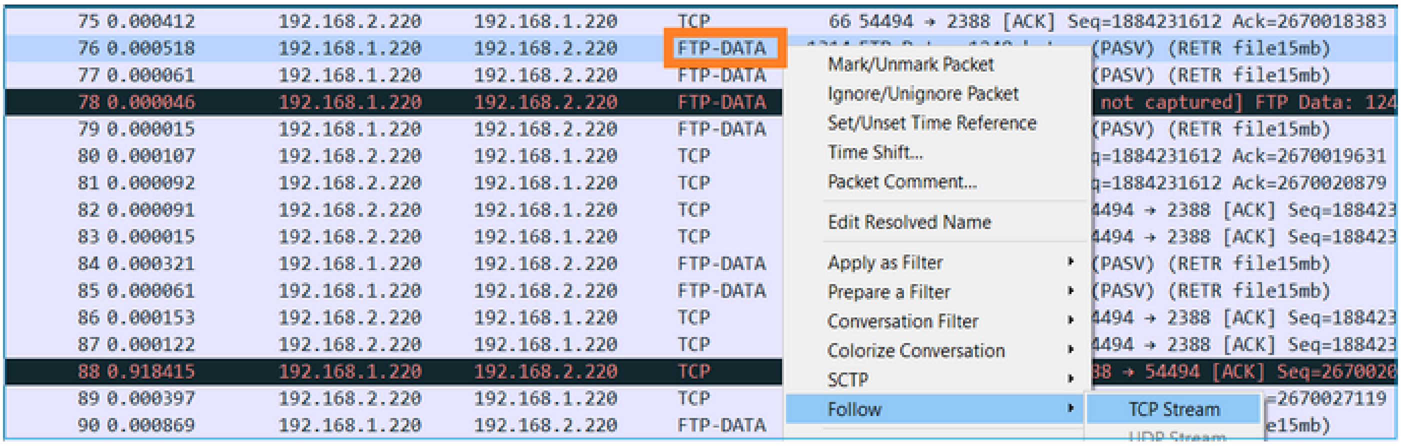
捕獲分析

在FTD LINA引擎上啟用擷取。

<#root>

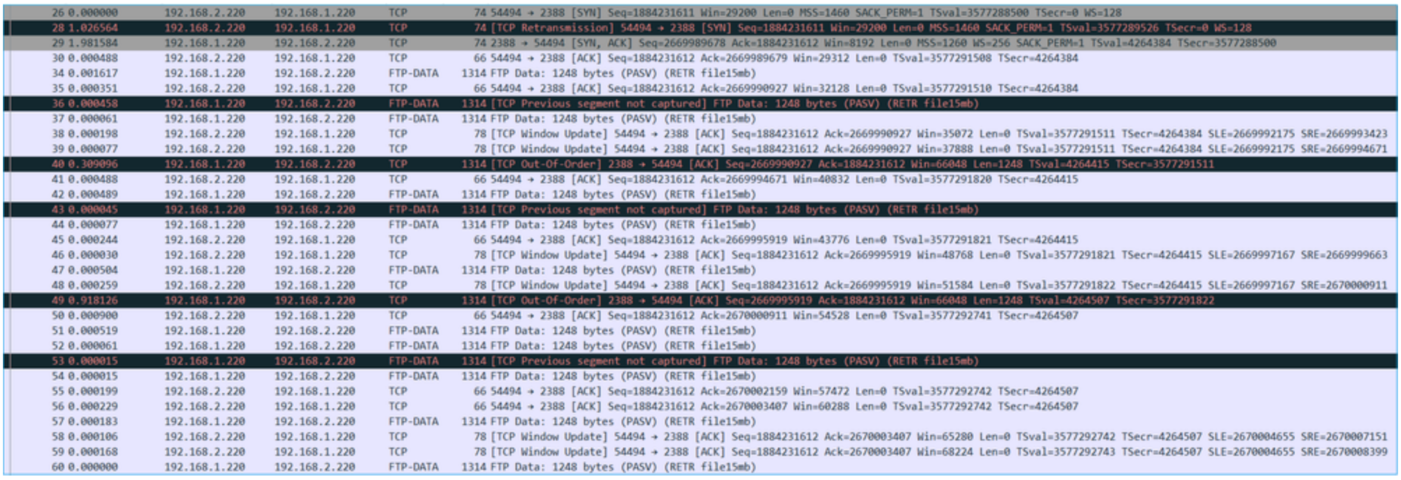
```
firepower#
capture CAPI type raw-data buffer 33554432 interface INSIDE match tcp host 192.168.2.220 host 192.168.1.
firepower#
cap CAPO type raw-data buffer 33554432 interface OUTSIDE match tcp host 192.168.2.220 host 192.168.1.220
```

選擇FTP-DATA封包，並依照FTD INSIDE capture(CAPI)上的FTP資料通道操作：



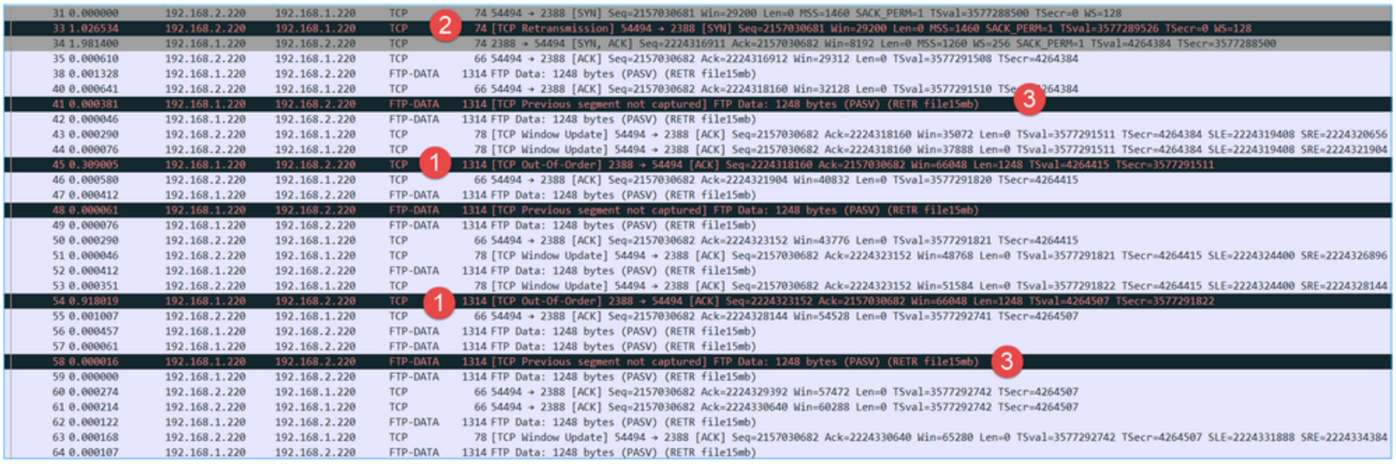
```
75 0.000412   192.168.2.220   192.168.1.220   TCP         66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670018383
76 0.000518   192.168.1.220   192.168.2.220   FTP-DATA    (PASV) (RETR file15mb)
77 0.000061   192.168.1.220   192.168.2.220   FTP-DATA    (PASV) (RETR file15mb)
78 0.000046   192.168.1.220   192.168.2.220   FTP-DATA    not captured] FTP Data: 124
79 0.000015   192.168.1.220   192.168.2.220   FTP-DATA    (PASV) (RETR file15mb)
80 0.000107   192.168.2.220   192.168.1.220   TCP         q=1884231612 Ack=2670019631
81 0.000092   192.168.2.220   192.168.1.220   TCP         q=1884231612 Ack=2670020879
82 0.000091   192.168.2.220   192.168.1.220   TCP         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         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         88 → 54494 [ACK] Seq=2670020
89 0.000397   192.168.2.220   192.168.1.220   TCP         =2670027119
90 0.000869   192.168.1.220   192.168.2.220   FTP-DATA    e15mb)
```

FTP-DATA流內容：



```
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 1.026564   192.168.2.220   192.168.1.220   TCP         74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
29 1.981584   192.168.1.220   192.168.2.220   TCP         74 2388 → 54494 [SYN, ACK] Seq=2669989678 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=2669989679 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
34 0.001617   192.168.1.220   192.168.2.220   FTP-DATA  1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
35 0.000351   192.168.2.220   192.168.1.220   TCP         66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 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=2669990927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=2669992175 SRE=2669993423
39 0.000077   192.168.2.220   192.168.1.220   TCP         78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=2669992175 SRE=2669994671
40 0.309096   192.168.1.220   192.168.2.220   TCP       1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669990927 Ack=1884231612 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
41 0.000488   192.168.2.220   192.168.1.220   TCP         66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669994671 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   192.168.1.220   192.168.2.220   FTP-DATA  1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
45 0.000244   192.168.2.220   192.168.1.220   TCP         66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 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=2669995919 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=2669997167 SRE=2669999663
47 0.000504   192.168.1.220   192.168.2.220   FTP-DATA  1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
48 0.000259   192.168.2.220   192.168.1.220   TCP         78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2669997167 SRE=2670000911
49 0.918126   192.168.1.220   192.168.2.220   TCP       1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669995919 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   192.168.1.220   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=2670004655 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捕獲內容：



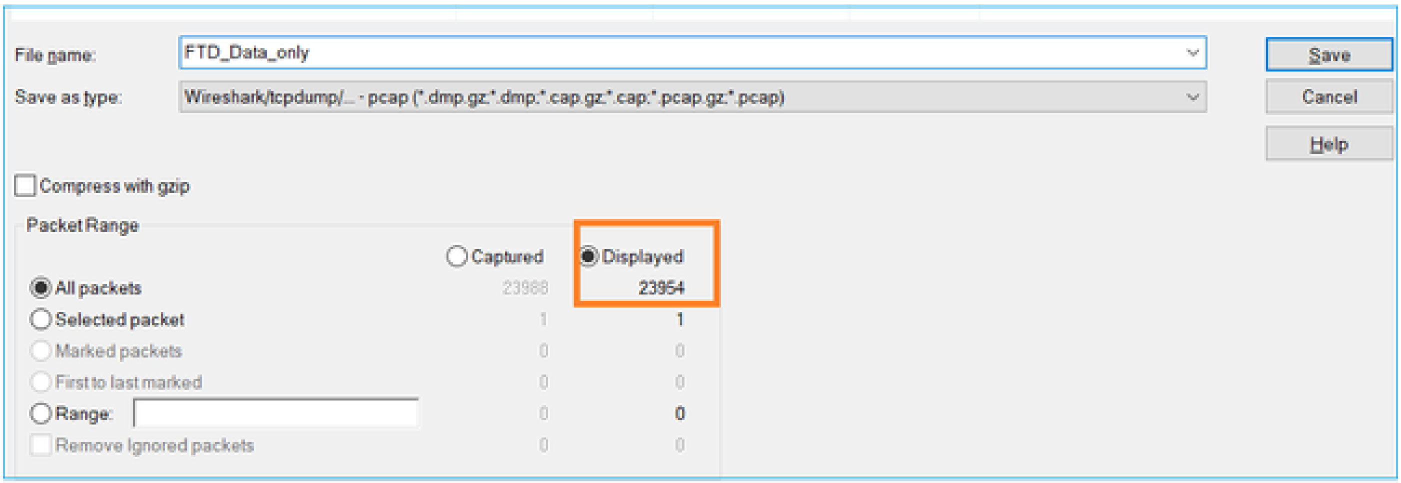
```
31 0.000000   192.168.2.220   192.168.1.220   TCP         74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
33 1.026534   192.168.2.220   192.168.1.220   TCP         74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 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=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
35 0.000610   192.168.2.220   192.168.1.220   TCP         66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
38 0.001328   192.168.1.220   192.168.2.220   FTP-DATA  1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
40 0.000641   192.168.2.220   192.168.1.220   TCP         66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSe  264384
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 0.000046   192.168.1.220   192.168.2.220   FTP-DATA  1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
43 0.000290   192.168.2.220   192.168.1.220   TCP         78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=2224319408 SRE=2224320656
44 0.000076   192.168.2.220   192.168.1.220   TCP         78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=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=2224321904 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   192.168.1.220   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=2224324400 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 0.000016   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)
```

重點：

1. 存在TCP亂序(OOO)封包。
2. 存在TCP重新傳輸。
3. 存在資料包丟失（丟棄的資料包）的指示。

提示：導航到File > Export Specified Packets時，儲存捕獲。然後僅儲存Displayed數據包範

圖



File name: FTD_Data_only

Save as type: Wireshark/tcpdump/... - pcap (*.dmp.gz;*.dmp;*.cap.gz;*.cap;*.pcap.gz;*.pcap)

□ Compress with gzip

Packet Range

	○ Captured	● Displayed
● All packets	23988	23954
○ Selected packet	1	1
○ Marked packets	0	0
○ First to last marked	0	0
○ Range:	0	0
□ Remove Ignored packets	0	0

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.確定資料包丟失位置。

在這種情況下，您必須採用同時捕獲並使用「分治法」來識別導致資料包丟失的網路段。從防火牆的角度來看，主要有3種情況：

1. 封包遺失是由防火牆本身造成的。
2. 封包遺失會在防火牆裝置的下游（從伺服器到使用者端的方向）引起。
3. 封包遺失是在防火牆裝置的上游（從使用者端到伺服器的方向）造成的。

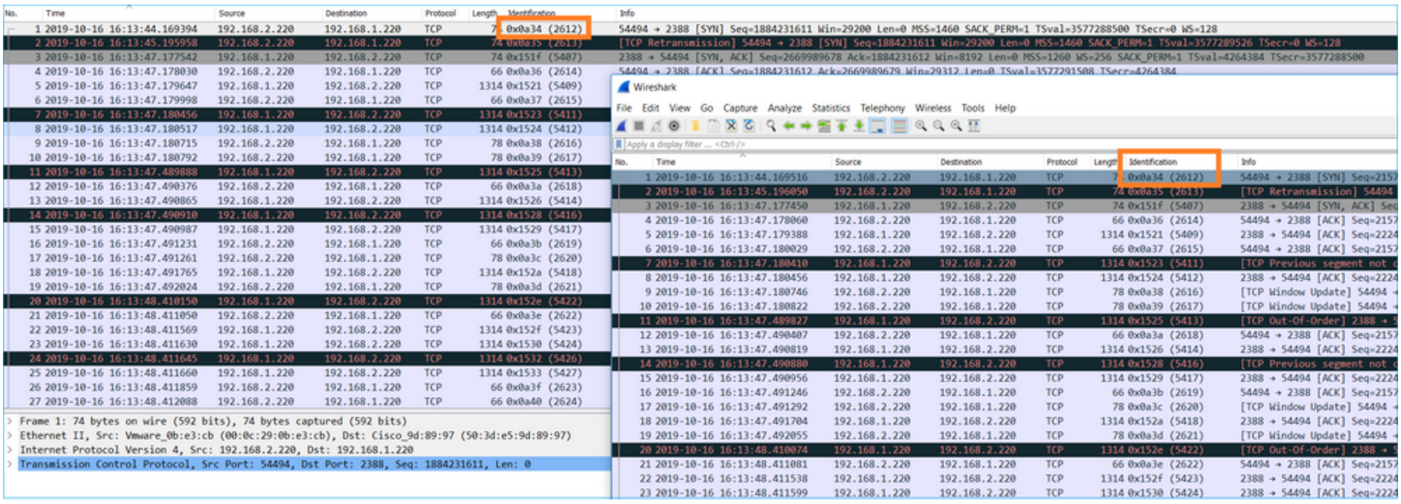
防火牆導致的資料包丟失：為了確定資料包丟失是否由防火牆引起，需要將入口捕獲與出口捕獲進行比較。有很多方法可以比較兩種不同的捕獲。本節演示一種執行此任務的方法。

比較2個擷取以識別封包遺失的程式

步驟 1.確保2個捕獲包含來自同一時間視窗的資料包。這表示一個擷取中一定沒有封包是在另一個擷取之前或之後擷取。有幾種方法可以做到這一點：

- 檢查第一個和最後一個封包IP識別(ID)值。
- 檢查第一個和最後一個資料包的時間戳值。

在此範例中，您可以看到每個擷取的第一個封包具有相同的IP ID值：



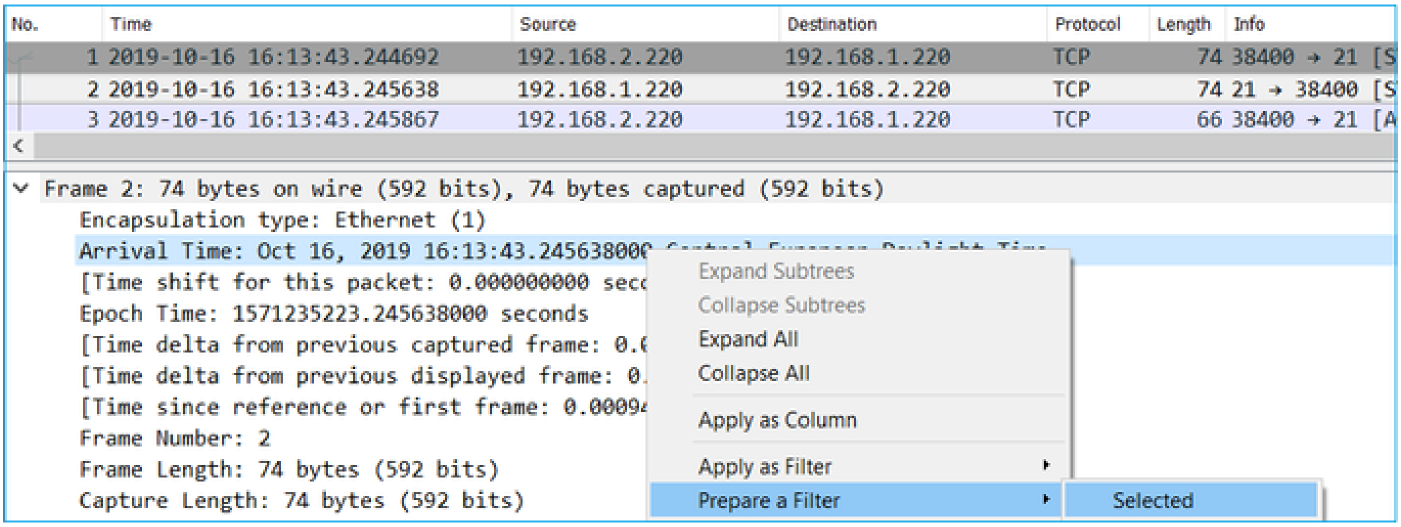
```
No.  Time                        Source         Destination    Protocol Length Identification Info
   1 2019-10-16 16:13:44.169394  192.168.2.220  192.168.1.220  TCP      7      0x0a34 (2612)  54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
   2 2019-10-16 16:13:45.195958  192.168.2.220  192.168.1.220  TCP      74     0x0a35 (2613)  [TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
   3 2019-10-16 16:13:47.177542  192.168.1.220  192.168.2.220  TCP      74     0x151f (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     0x0a36 (2614)  54494 → 2388 [ACK] Seq=1884231612 Ack=2669989679 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
   5 2019-10-16 16:13:47.179647  192.168.1.220  192.168.2.220  TCP      1314   0x1521 (5409)
   6 2019-10-16 16:13:47.179998  192.168.2.220  192.168.1.220  TCP      66     0x0a37 (2615)
   7 2019-10-16 16:13:47.180456  192.168.1.220  192.168.2.220  TCP      1314   0x1523 (5411)
   8 2019-10-16 16:13:47.180517  192.168.1.220  192.168.2.220  TCP      1314   0x1524 (5412)
   9 2019-10-16 16:13:47.180715  192.168.2.220  192.168.1.220  TCP      78     0x0a38 (2616)
  10 2019-10-16 16:13:47.180792  192.168.2.220  192.168.1.220  TCP      78     0x0a39 (2617)
  11 2019-10-16 16:13:47.489888  192.168.1.220  192.168.2.220  TCP      1314   0x1525 (5413)
  12 2019-10-16 16:13:47.490376  192.168.2.220  192.168.1.220  TCP      66     0x0a3a (2618)
  13 2019-10-16 16:13:47.490865  192.168.1.220  192.168.2.220  TCP      1314   0x1526 (5414)
  14 2019-10-16 16:13:47.490910  192.168.1.220  192.168.2.220  TCP      1314   0x1528 (5416)
  15 2019-10-16 16:13:47.490987  192.168.1.220  192.168.2.220  TCP      1314   0x1529 (5417)
  16 2019-10-16 16:13:47.491231  192.168.2.220  192.168.1.220  TCP      66     0x0a3b (2619)
  17 2019-10-16 16:13:47.491261  192.168.2.220  192.168.1.220  TCP      78     0x0a3c (2620)
  18 2019-10-16 16:13:47.491765  192.168.1.220  192.168.2.220  TCP      1314   0x152a (5418)
  19 2019-10-16 16:13:47.492024  192.168.2.220  192.168.1.220  TCP      78     0x0a3d (2621)
  20 2019-10-16 16:13:48.410150  192.168.1.220  192.168.2.220  TCP      1314   0x152e (5422)
  21 2019-10-16 16:13:48.411050  192.168.2.220  192.168.1.220  TCP      66     0x0a3e (2622)
  22 2019-10-16 16:13:48.411569  192.168.1.220  192.168.2.220  TCP      1314   0x152f (5423)
  23 2019-10-16 16:13:48.411630  192.168.1.220  192.168.2.220  TCP      1314   0x1530 (5424)
  24 2019-10-16 16:13:48.411645  192.168.1.220  192.168.2.220  TCP      1314   0x1532 (5426)
  25 2019-10-16 16:13:48.411660  192.168.1.220  192.168.2.220  TCP      1314   0x1533 (5427)
  26 2019-10-16 16:13:48.411859  192.168.2.220  192.168.1.220  TCP      66     0x0a3f (2623)
  27 2019-10-16 16:13:48.412088  192.168.2.220  192.168.1.220  TCP      66     0x0a40 (2624)

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: Vmware_0b:e3:cb (00:0c:29:0b:e3:cb), Dst: Cisco_9d:89:97 (50:3d:e5:9d:89:97)
> Internet Protocol Version 4, Src: 192.168.2.220, Dst: 192.168.1.220
> Transmission Control Protocol, Src Port: 54494, Dst Port: 2388, Seq: 1884231611, Len: 0
```

Wireshark
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time                        Source         Destination    Protocol Length Identification Info
   1 2019-10-16 16:13:44.169516  192.168.2.220  192.168.1.220  TCP      7      0x0a34 (2612)  54494 → 2388 [SYN] Seq=2157
   2 2019-10-16 16:13:45.196050  192.168.2.220  192.168.1.220  TCP      74     0x0a35 (2613)  [TCP Retransmission] 54494
   3 2019-10-16 16:13:47.177450  192.168.1.220  192.168.2.220  TCP      74     0x151f (5407)  2388 → 54494 [SYN, ACK] Seq
   4 2019-10-16 16:13:47.178060  192.168.2.220  192.168.1.220  TCP      66     0x0a36 (2614)  54494 → 2388 [ACK] Seq=2157
   5 2019-10-16 16:13:47.179388  192.168.1.220  192.168.2.220  TCP      1314   0x1521 (5409)  2388 → 54494 [ACK] Seq=2224
   6 2019-10-16 16:13:47.180029  192.168.2.220  192.168.1.220  TCP      66     0x0a37 (2615)  54494 → 2388 [ACK] Seq=2157
   7 2019-10-16 16:13:47.180410  192.168.1.220  192.168.2.220  TCP      1314   0x1523 (5411)  [TCP Previous segment not c
   8 2019-10-16 16:13:47.180456  192.168.1.220  192.168.2.220  TCP      1314   0x1524 (5412)  2388 → 54494 [ACK] Seq=2224
   9 2019-10-16 16:13:47.180746  192.168.2.220  192.168.1.220  TCP      78     0x0a38 (2616)  [TCP Window Update] 54494 →
  10 2019-10-16 16:13:47.180822  192.168.2.220  192.168.1.220  TCP      78     0x0a39 (2617)  [TCP Window Update] 54494 →
  11 2019-10-16 16:13:47.489827  192.168.1.220  192.168.2.220  TCP      1314   0x1525 (5413)  [TCP Out-Of-Order] 2388 → 5
  12 2019-10-16 16:13:47.490407  192.168.2.220  192.168.1.220  TCP      66     0x0a3a (2618)  54494 → 2388 [ACK] Seq=2157
  13 2019-10-16 16:13:47.490819  192.168.1.220  192.168.2.220  TCP      1314   0x1526 (5414)  2388 → 54494 [ACK] Seq=2224
  14 2019-10-16 16:13:47.490880  192.168.1.220  192.168.2.220  TCP      1314   0x1528 (5416)  [TCP Previous segment not c
  15 2019-10-16 16:13:47.490956  192.168.1.220  192.168.2.220  TCP      1314   0x1529 (5417)  2388 → 54494 [ACK] Seq=2224
  16 2019-10-16 16:13:47.491246  192.168.2.220  192.168.1.220  TCP      66     0x0a3b (2619)  54494 → 2388 [ACK] Seq=2157
  17 2019-10-16 16:13:47.491292  192.168.2.220  192.168.1.220  TCP      78     0x0a3c (2620)  [TCP Window Update] 54494 →
  18 2019-10-16 16:13:47.491704  192.168.1.220  192.168.2.220  TCP      1314   0x152a (5418)  2388 → 54494 [ACK] Seq=2224
  19 2019-10-16 16:13:47.492055  192.168.2.220  192.168.1.220  TCP      78     0x0a3d (2621)  [TCP Window Update] 54494 →
  20 2019-10-16 16:13:48.410074  192.168.1.220  192.168.2.220  TCP      1314   0x152e (5422)  [TCP Out-Of-Order] 2388 → 5
  21 2019-10-16 16:13:48.411081  192.168.2.220  192.168.1.220  TCP      66     0x0a3e (2622)  54494 → 2388 [ACK] Seq=2157
  22 2019-10-16 16:13:48.411538  192.168.1.220  192.168.2.220  TCP      1314   0x152f (5423)  2388 → 54494 [ACK] Seq=2224
  23 2019-10-16 16:13:48.411599  192.168.1.220  192.168.2.220  TCP      1314   0x1530 (5424)  2388 → 54494 [ACK] Seq=2224
```

如果它們不同，那麼：

1. 比較每個捕獲的第一個資料包的時間戳。
2. 從具有最新時間戳的捕獲獲取過濾器，從中將Timestamp過濾器從==更改為>=（第一個資料包）和<=（最後一個資料包）更改，例如：



```
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

v 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 Central European Daylight Time
    [Time shift for this packet: 0.000000000 seco
    Epoch Time: 1571235223.245638000 seconds
    [Time delta from previous captured frame: 0.0
    [Time delta from previous displayed frame: 0.
    [Time since reference or first frame: 0.00094
    Frame Number: 2
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
```

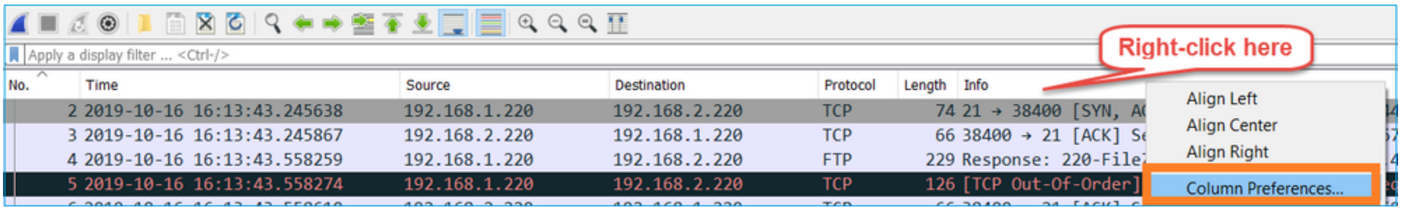
(frame.time >= "2019年10月16日16:13:43.244692000")和&(frame.time <= "2019年10月16日16:20:21.785130000")

3.將指定的資料包匯出到新捕獲，選擇檔案>匯出指定的資料包，然後儲存顯示的資料包。此時，兩個捕獲都必須包含覆蓋同一時間視窗的資料包。現在，您可以開始比較2個捕獲。

步驟 2.指定用於比較2個捕獲的資料包欄位。可以使用的欄位示例：

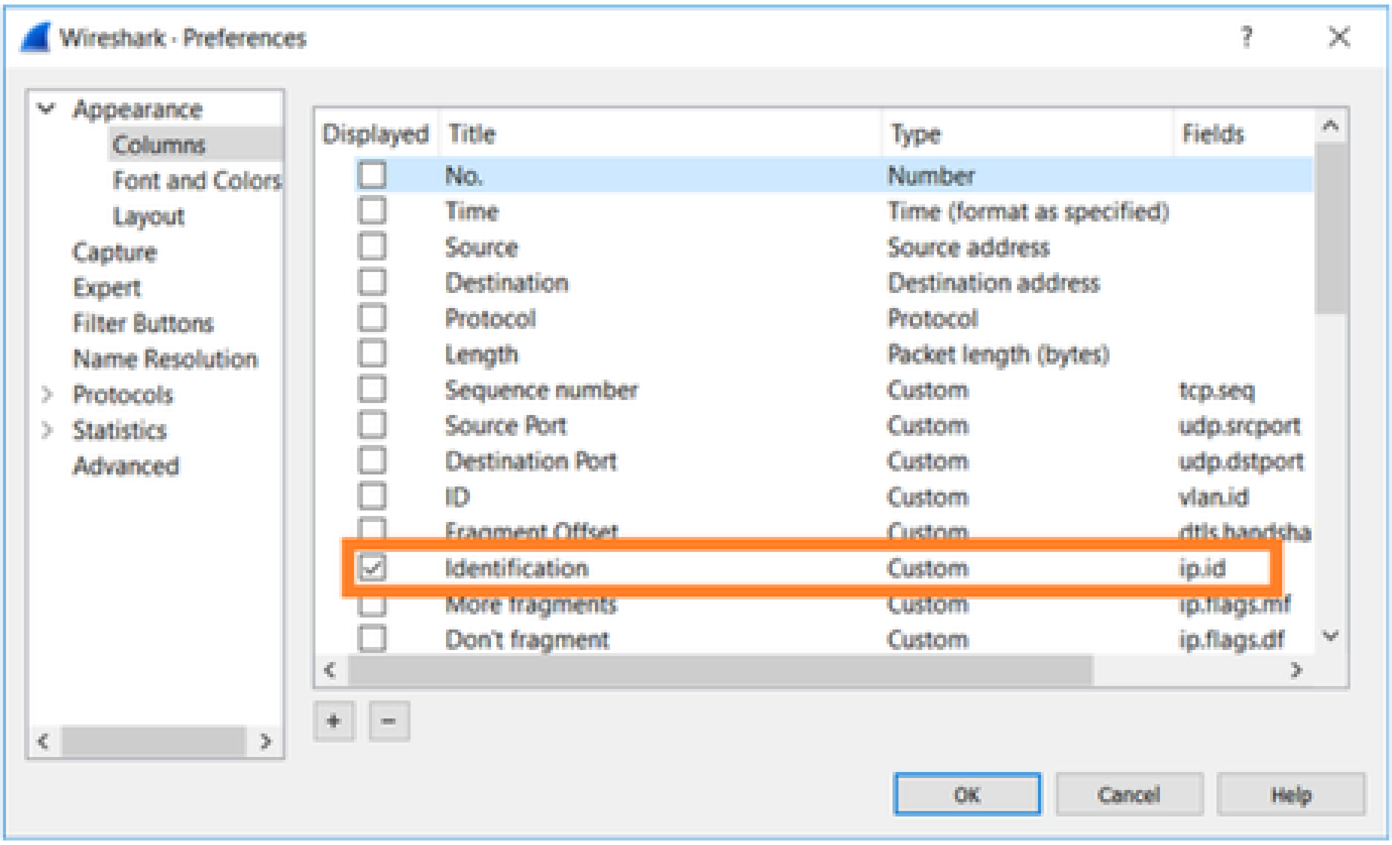
- IP識別
- RTP序列號
- ICMP序列號

建立每個捕獲的文本版本，其中包含您在步驟1中指定的每個資料包的欄位。為此，請僅保留感興趣的列，例如，如果要根據IP標識比較資料包，請修改捕獲，如下圖所示。



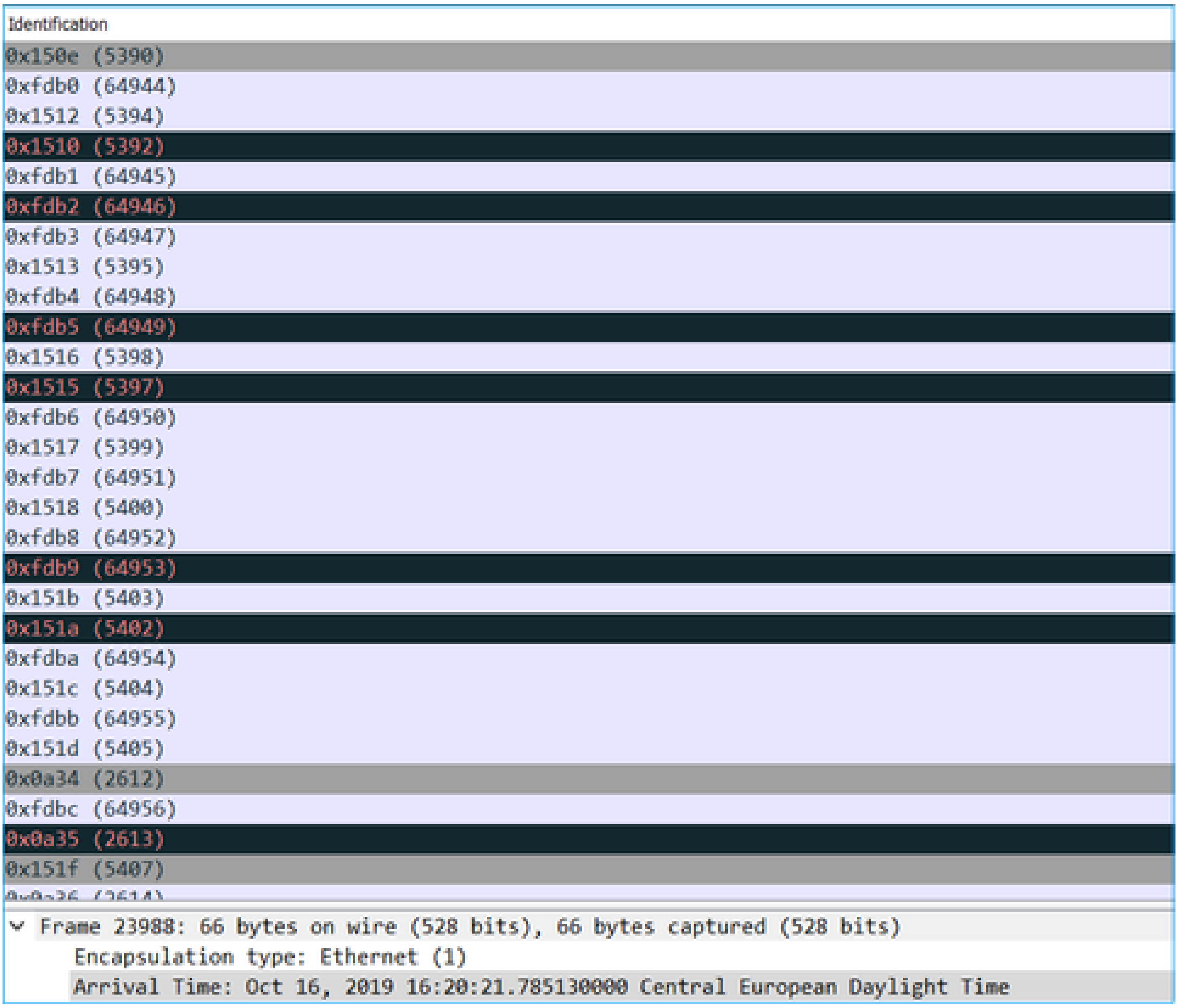
Apply a display filter ... <Ctrl-/>

```
No.   Time                          Source          Destination     Protocol  Length  Info
    2 2019-10-16 16:13:43.245638    192.168.1.220   192.168.2.220   TCP           74  21 → 38400 [SYN, AC
    3 2019-10-16 16:13:43.245867    192.168.2.220   192.168.1.220   TCP           66  38400 → 21 [ACK] Se
    4 2019-10-16 16:13:43.558259    192.168.1.220   192.168.2.220   FTP          229  Response: 220-File
    5 2019-10-16 16:13:43.558274    192.168.1.220   192.168.2.220   TCP          126  [TCP Out-Of-Order]
```



Displayed	Title	Type	Fields
□	No.	Number	
□	Time	Time (format as specified)	
□	Source	Source address	
□	Destination	Destination address	
□	Protocol	Protocol	
□	Length	Packet length (bytes)	
□	Sequence number	Custom	tcp.seq
□	Source Port	Custom	udp.srcport
□	Destination Port	Custom	udp.dstport
□	ID	Custom	vlan.id
□	Fragment Offset	Custom	dtls.handsha
☑	Identification	Custom	ip.id
□	More fragments	Custom	ip.flags.mf
□	Don't fragment	Custom	ip.flags.df

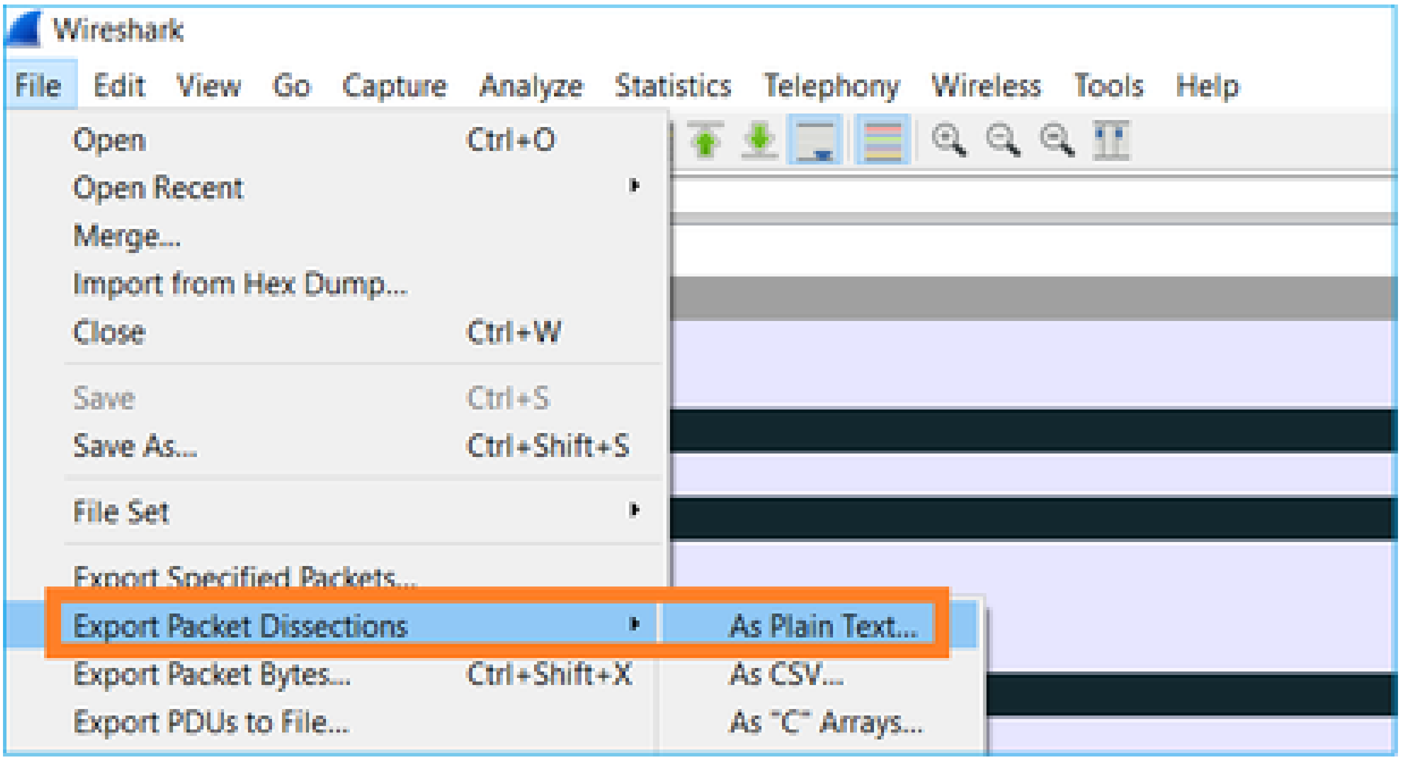
結果是：



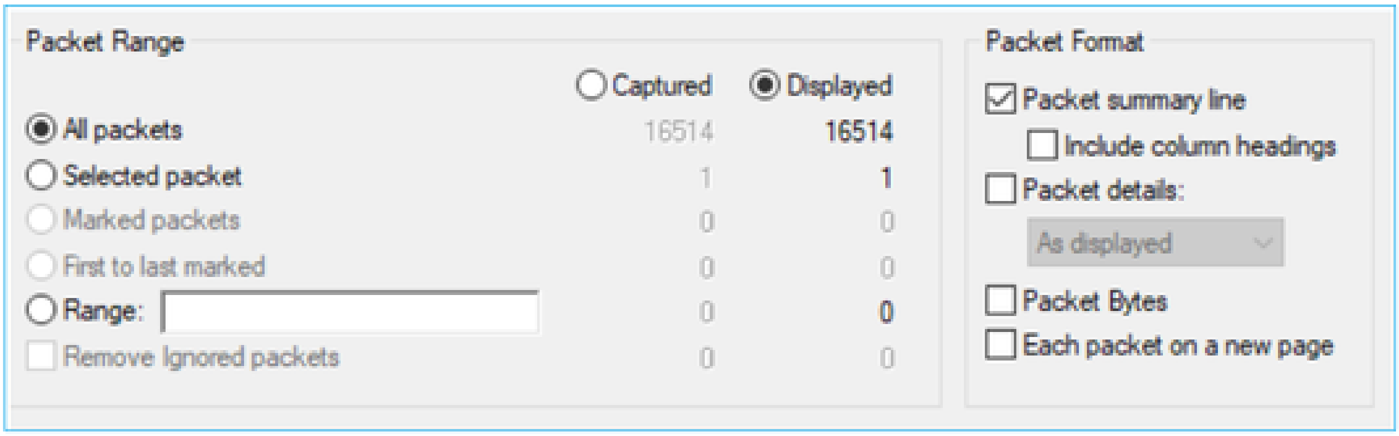
```
Identification
0x150e (5390)
0xfdb0 (64944)
0x1512 (5394)
0x1510 (5392)
0xfdb1 (64945)
0xfdb2 (64946)
0xfdb3 (64947)
0x1513 (5395)
0xfdb4 (64948)
0xfdb5 (64949)
0x1516 (5398)
0x1515 (5397)
0xfdb6 (64950)
0x1517 (5399)
0xfdb7 (64951)
0x1518 (5400)
0xfdb8 (64952)
0xfdb9 (64953)
0x151b (5403)
0x151a (5402)
0xfdba (64954)
0x151c (5404)
0xfdbb (64955)
0x151d (5405)
0x0a34 (2612)
0xfdbc (64956)
0x0a35 (2613)
0x151f (5407)
0x0a36 (2614)
```

```
v Frame 23988: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
      Encapsulation type: Ethernet (1)
      Arrival Time: Oct 16, 2019 16:20:21.785130000 Central European Daylight Time
```

步驟 3.建立擷取的文字版本(「檔案」>「匯出封包分段」>「以純文本......」)，如下圖所示：



取消選中Include column headings和Packet details選項，以僅匯出所顯示欄位的值，如下圖所示：



Packet Range

	○ Captured	● Displayed
● All packets	16514	16514
○ Selected packet	1	1
○ Marked packets	0	0
○ First to last marked	0	0
○ Range:	0	0
□ Remove Ignored packets	0	0

Packet Format

☑ Packet summary line

□ Include column headings

□ Packet details:

As displayed

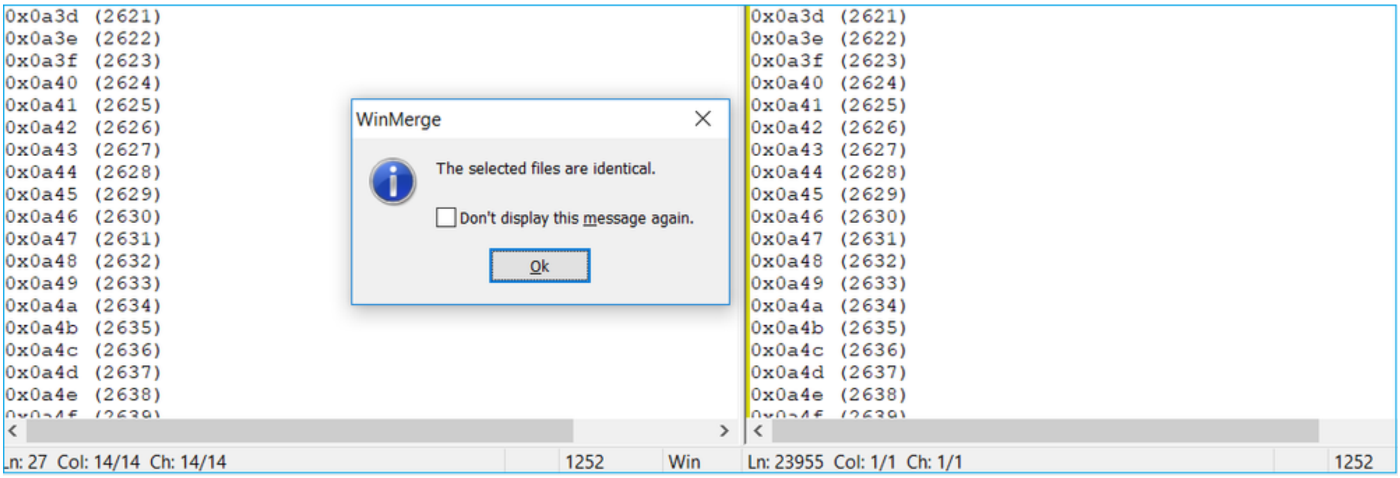
□ Packet Bytes

□ Each packet on a new page

步驟 4.對檔案中的資料包進行排序。您可以使用Linux sort命令執行以下操作：

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

步驟 5.使用文本比較工具（例如WinMerge）或Linux diff命令查詢2個捕獲之間的差異。



```
0x0a3d (2621)                                    0x0a3d (2621)
0x0a3e (2622)                                    0x0a3e (2622)
0x0a3f (2623)                                    0x0a3f (2623)
0x0a40 (2624)                                    0x0a40 (2624)
0x0a41 (2625)                                    0x0a41 (2625)
0x0a42 (2626)                                    0x0a42 (2626)
0x0a43 (2627)                                    0x0a43 (2627)
0x0a44 (2628)                                    0x0a44 (2628)
0x0a45 (2629)                                    0x0a45 (2629)
0x0a46 (2630)                                    0x0a46 (2630)
0x0a47 (2631)                                    0x0a47 (2631)
0x0a48 (2632)                                    0x0a48 (2632)
0x0a49 (2633)                                    0x0a49 (2633)
0x0a4a (2634)                                    0x0a4a (2634)
0x0a4b (2635)                                    0x0a4b (2635)
0x0a4c (2636)                                    0x0a4c (2636)
0x0a4d (2637)                                    0x0a4d (2637)
0x0a4e (2638)                                    0x0a4e (2638)
0x0a4f (2639)                                    0x0a4f (2639)
Ln: 27  Col: 14/14  Ch: 14/14     1252    Win    Ln: 23955  Col: 1/1  Ch: 1/1     1252
```

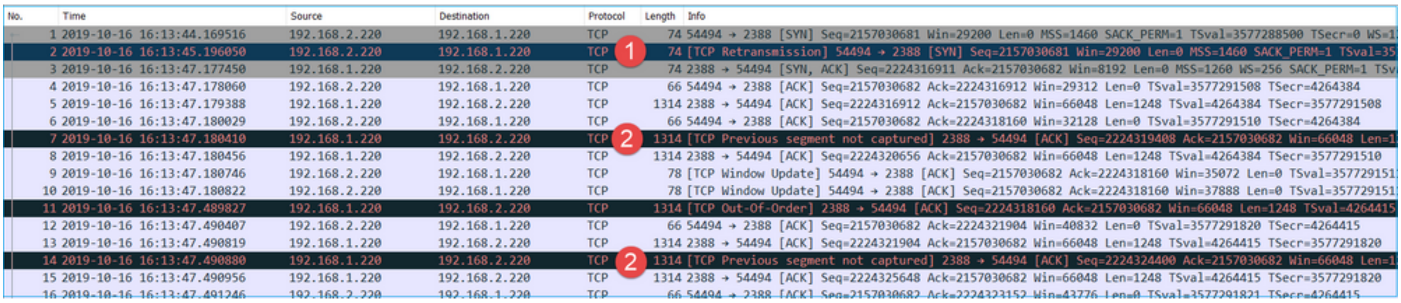
WinMerge

The selected files are identical.

□ Don't display this message again.

在這種情況下，用於FTP資料流量的CAPI和CAPO捕獲完全相同。這證明封包遺失不是防火牆造成的。

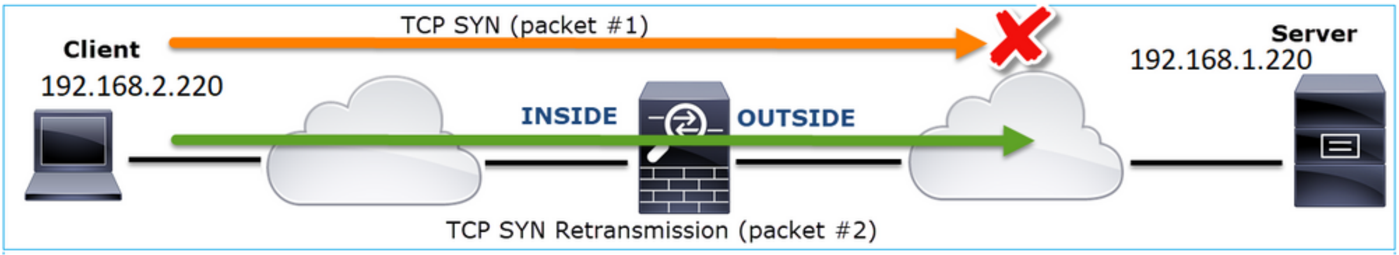
確定上游/下游資料包丟失。



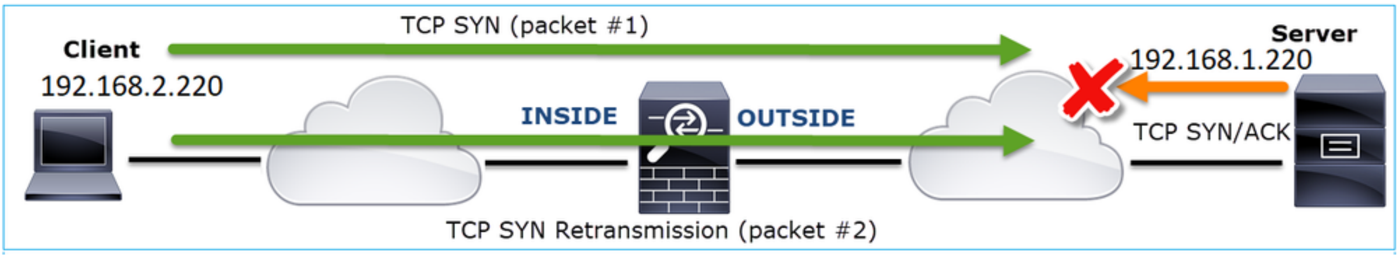
```
No.  Time                        Source         Destination    Protocol Length Info
   1 2019-10-16 16:13:44.169516  192.168.2.220  192.168.1.220  TCP        74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=1
   2 2019-10-16 16:13:45.196050  192.168.2.220  192.168.1.220  TCP        74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=35
   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 TSv
   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=1
   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=357729151
  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=357729151
  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
  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=1
  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
```

重點：

1.此資料包是TCP重傳資料包。具體而言，它是從客戶端傳送到伺服器的TCP SYN資料包，用於被動模式下的FTP資料。由於使用者端重新傳送封包，且您可以看到初始SYN(封包#1)，因此封包在防火牆的上游已遺失。

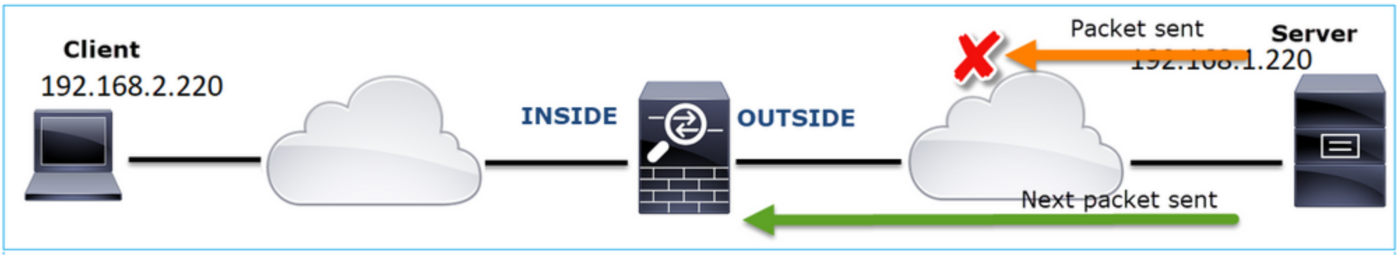


在這種情況下，有可能是SYN封包到達伺服器，但SYN/ACK封包在傳回時遺失：



2.從伺服器發出一個資料包，Wireshark發現未看到/捕獲上一個資料段。由於未捕獲的資料包從伺

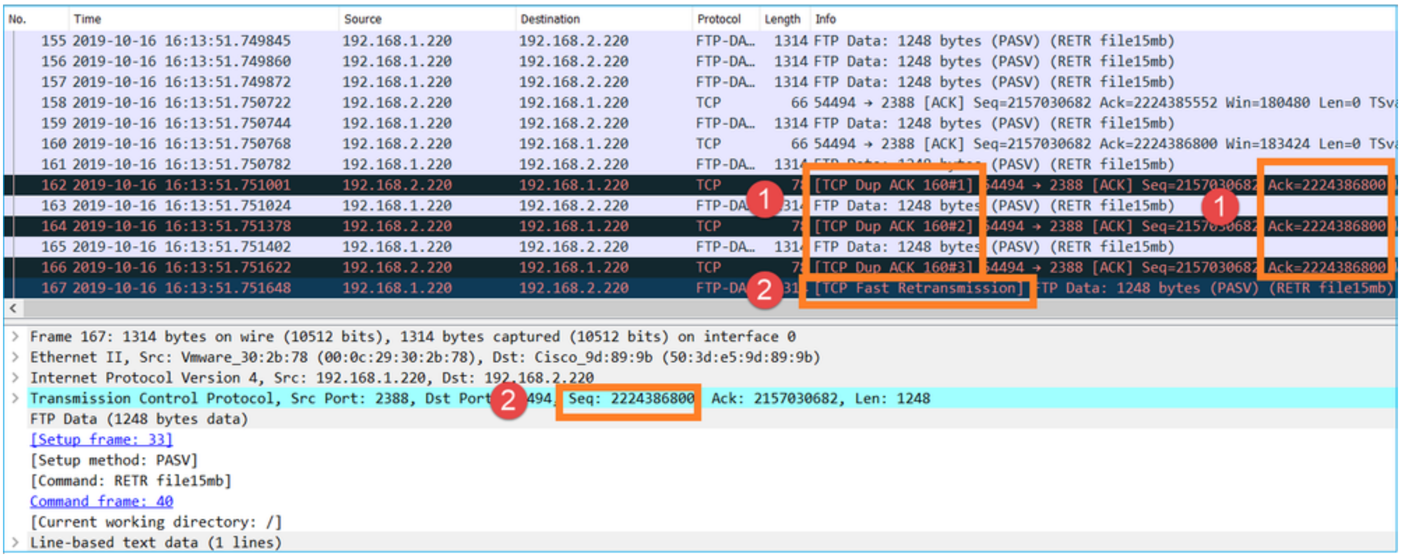
服器傳送到客戶端，而且在防火牆捕獲中看不到此資料包，這意味著該資料包在伺服器和防火牆之間丟失。



這表示FTP伺服器和防火牆之間發生封包遺失。

行動2.獲取其他捕獲。

在終端處獲取其他捕獲和捕獲。嘗試應用分治法進一步隔離導致資料包丟失的有問題的資料段。



```
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 TSva
 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 TSva
 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        7  [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   131  FTP Data: 1248 bytes (PASV) (RETR file15mb)
 164 2019-10-16 16:13:51.751378  192.168.2.220  192.168.1.220  TCP        7  [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…  131  FTP Data: 1248 bytes (PASV) (RETR file15mb)
 166 2019-10-16 16:13:51.751622  192.168.2.220  192.168.1.220  TCP        7  [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   31   [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)
```

重點：

1. 接收器（本例中為FTP客戶端）跟蹤傳入的TCP序列號。如果檢測到資料包丟失（已跳過預期的序列號），則生成帶有ACK='已跳過預期序列號'的ACK資料包。在此示例中，Ack=2224386800。
2. Dup ACK會觸發TCP快速重新傳輸（收到重複的ACK後，將在20毫秒內重新傳輸）。

重複ACK是什麼意思？

- 有幾個ACK重複，但沒有實際重新傳輸，這表明更有可能有資料包到達順序混亂。
- 重複的ACK和實際的重新傳輸表明存在一定程度的資料包丟失。

行動3.計算傳輸資料包的防火牆處理時間。

將相同的捕獲應用於2個不同的介面：

<#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
```

匯出捕獲檢查入口資料包與出口資料包之間的時間差

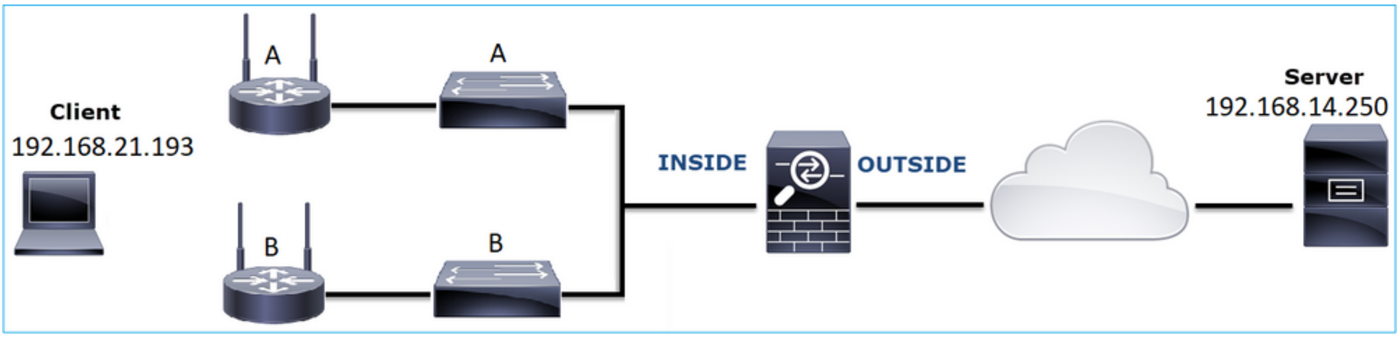
案例7.TCP連線問題（封包損毀）

問題描述：

無線客戶端(192.168.21.193)嘗試連線到目標伺服器(192.168.14.250 - HTTP)，並且有兩種不同的情況：

- 當客戶端連線到接入點(AP)的「A」時，HTTP連線不起作用。
- 當客戶端連線到接入點(AP)「B」時，HTTP連線會正常工作。

下圖顯示拓撲：



受影響的流：

源IP:192.168.21.193

Dst IP:192.168.14.250

協定：TCP 80

捕獲分析

在FTD LINA引擎上啟用擷取：

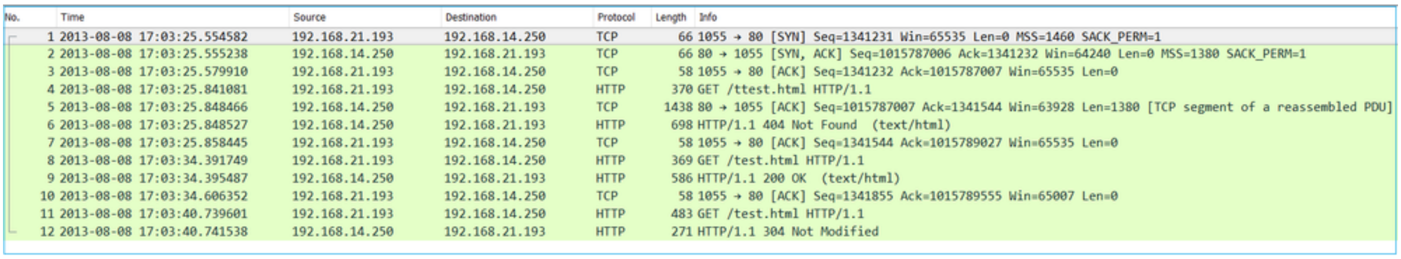
<#root>

```
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
```

捕獲 — 功能場景：

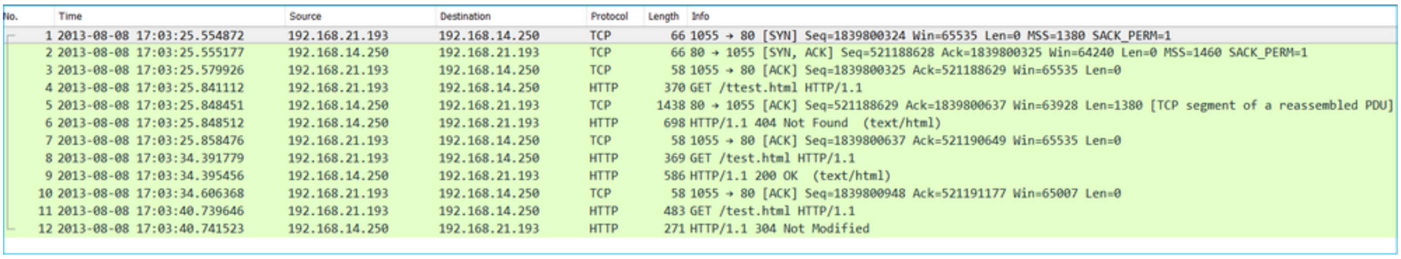
作為基準，使用已知良好方案中的捕獲始終非常有用。

此圖顯示NGFW INSIDE介面上的擷取



```
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
```

此圖顯示在NGFW OUTSIDE介面上捕獲的流量。



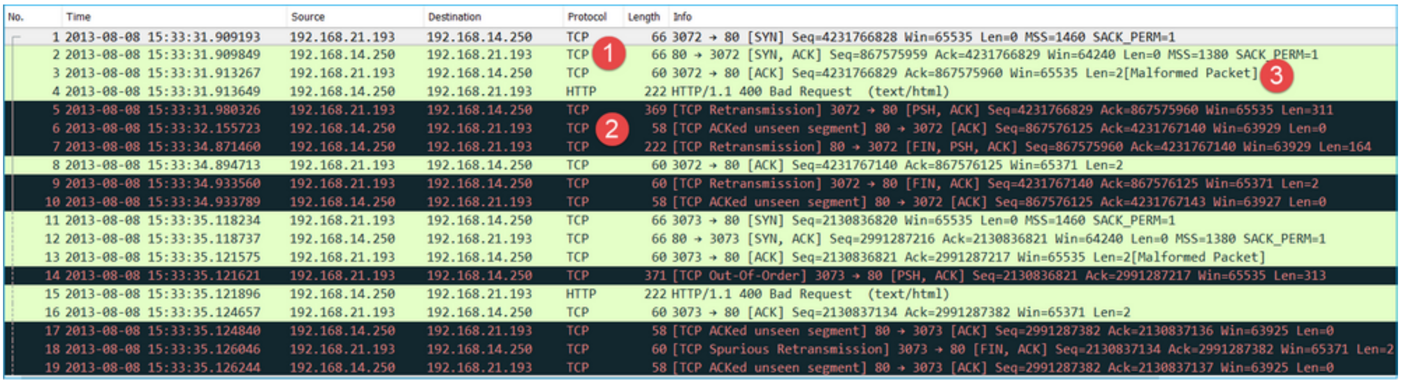
```
No.  Time                        Source           Destination      Protocol  Length  Info
   1 2013-08-08 17:03:25.554872  192.168.21.193   192.168.14.250   TCP         66 1055 → 80 [SYN] Seq=1839800324 Win=65535 Len=0 MSS=1380 SACK_PERM=1
   2 2013-08-08 17:03:25.555177  192.168.14.250   192.168.21.193   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
```

重點：

1. 2個捕獲幾乎完全相同（考慮ISN隨機化）。
2. 沒有資料包丟失的跡象。
3. 沒有亂序(OOO)封包
4. 有3個HTTP GET請求。第一個獲得404「未找到」，第二個獲得200「正常」，第三個獲得304「未修改」重定向消息。

捕獲 — 已知故障場景：

輸入擷取(CAPI)內容。

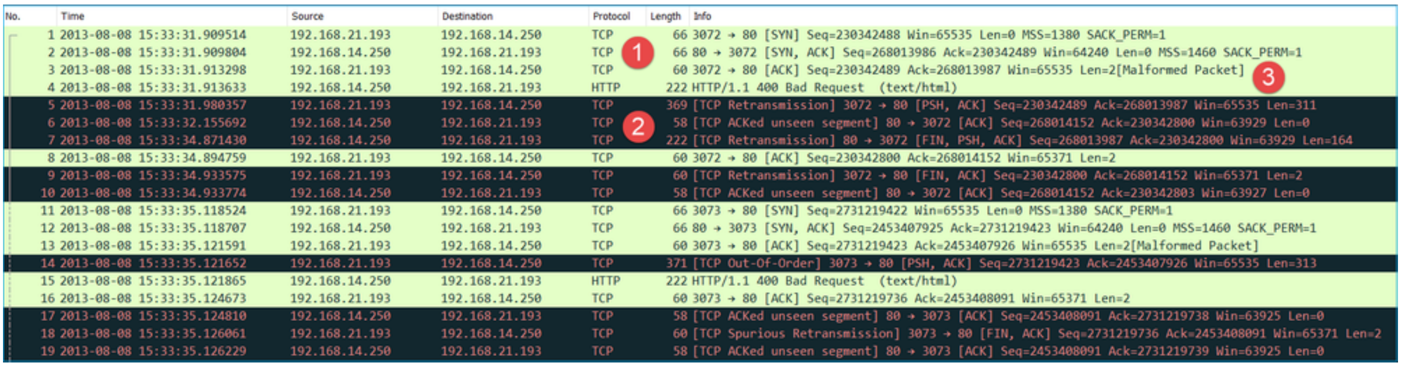


No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909193	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 15:33:31.909849	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=867575959 Ack=4231766829 Win=64240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

重點：

1. 有一個TCP三次握手。
2. 存在TCP重新傳輸和資料包丟失指示。
3. 有一個封包(TCP ACK)被Wireshark識別為Malformed。

此圖顯示輸出擷取(CAPO)內容。



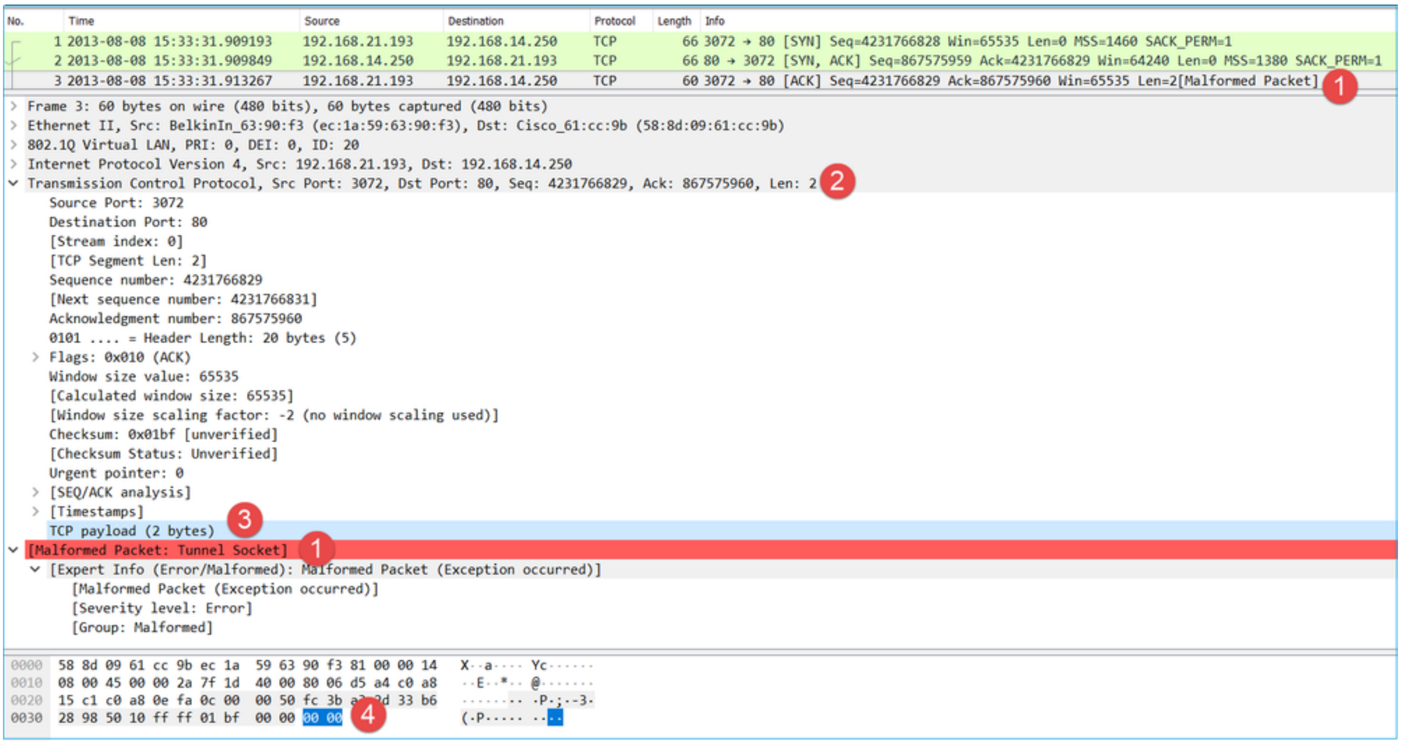
No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909514	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=230342488 Win=65535 Len=0 MSS=1380 SACK_PERM=1
2	2013-08-08 15:33:31.909804	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=268013986 Ack=230342489 Win=64240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

重點：

2個捕獲幾乎完全相同（考慮ISN隨機化）：

1. 有一個TCP三次握手。
2. 存在TCP重新傳輸和資料包丟失指示。
3. 有一個封包(TCP ACK)被Wireshark識別為Malformed。

檢查格式錯誤的資料包：



```
No.  Time                         Source          Destination     Protocol  Length  Info
   1 2013-08-08 15:33:31.909193   192.168.21.193  192.168.14.250  TCP           66 3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=1460 SACK_PERM=1
   2 2013-08-08 15:33:31.909849   192.168.14.250  192.168.21.193  TCP           66 80 → 3072 [SYN, ACK] Seq=867575959 Ack=4231766829 Win=64240 Len=0 MSS=1380 SACK_PERM=1
   3 2013-08-08 15:33:31.913267   192.168.21.193  192.168.14.250  TCP           60 3072 → 80 [ACK] Seq=4231766829 Ack=867575960 Win=65535 Len=2[Malformed Packet]
> Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: BelkinIn_63:90:f3 (ec:1a:59:63:90:f3), Dst: Cisco_61:cc:9b (58:8d:09:61:cc:9b)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
> Internet Protocol Version 4, Src: 192.168.21.193, Dst: 192.168.14.250
v Transmission Control Protocol, Src Port: 3072, Dst Port: 80, Seq: 4231766829, Ack: 867575960, Len: 2
    Source Port: 3072
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 2]
    Sequence number: 4231766829
    [Next sequence number: 4231766831]
    Acknowledgment number: 867575960
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 65535
    [Calculated window size: 65535]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x01bf [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (2 bytes)
v [Malformed Packet: Tunnel Socket]
  v [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
      [Malformed Packet (Exception occurred)]
      [Severity level: Error]
      [Group: Malformed]

0000  58 8d 09 61 cc 9b ec 1a  59 63 90 f3 81 00 00 14   X··a···· Yc······
0010  08 00 45 00 00 2a 7f 1d  40 00 80 06 d5 a4 c0 a8   ··E··*·· @·······
0020  15 c1 c0 a8 0e fa 0c 00  00 50 fc 3b a3 2d 33 b6   ········ ·P·;·-3·
0030  28 98 50 10 ff ff 01 bf  00 00 00 00               (·P····· ····
```

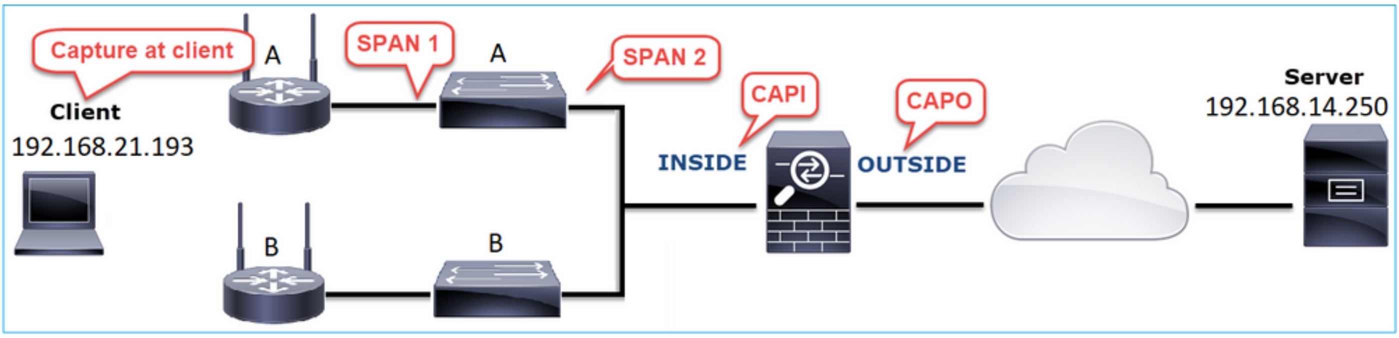
重點：

1. 資料包被識別為Wireshark的Malformed。
2. 長度為2個位元組。
3. 有2個位元組的TCP負載。
4. 負載是額外的4個0(00 00)。

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.獲取其他捕獲。包括終端上的捕獲，如果可能，請嘗試應用分治法來隔離資料包損壞的來源，例如：

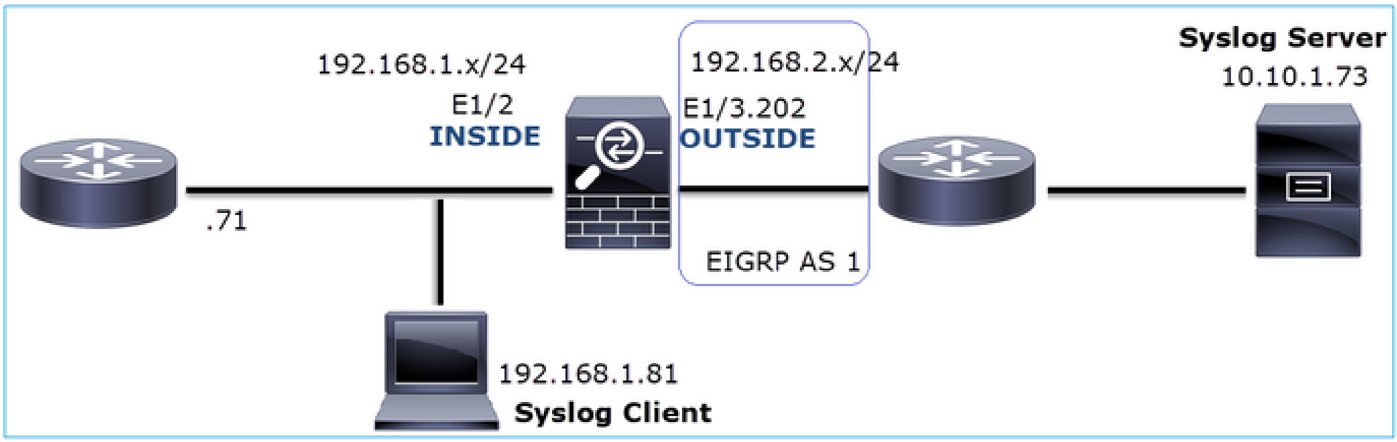


在這種情況下，交換器「A」介面驅動程式增加了額外的2位元組，解決方式是取代導致損毀的交換器。

案例8.UDP連線問題（缺少資料包）

問題描述：在目標Syslog伺服器上看不到系統日誌(UDP 514)消息。

下圖顯示拓撲：



受影響的流：

源IP:192.168.1.81

Dst IP:10.10.1.73

協定：UDP 514

捕獲分析

在FTD LINA引擎上啟用擷取：

<#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擷取show no packets:

<#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
```

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.檢查FTD連線表。

要檢查特定連線，可以使用以下語法：

<#root>

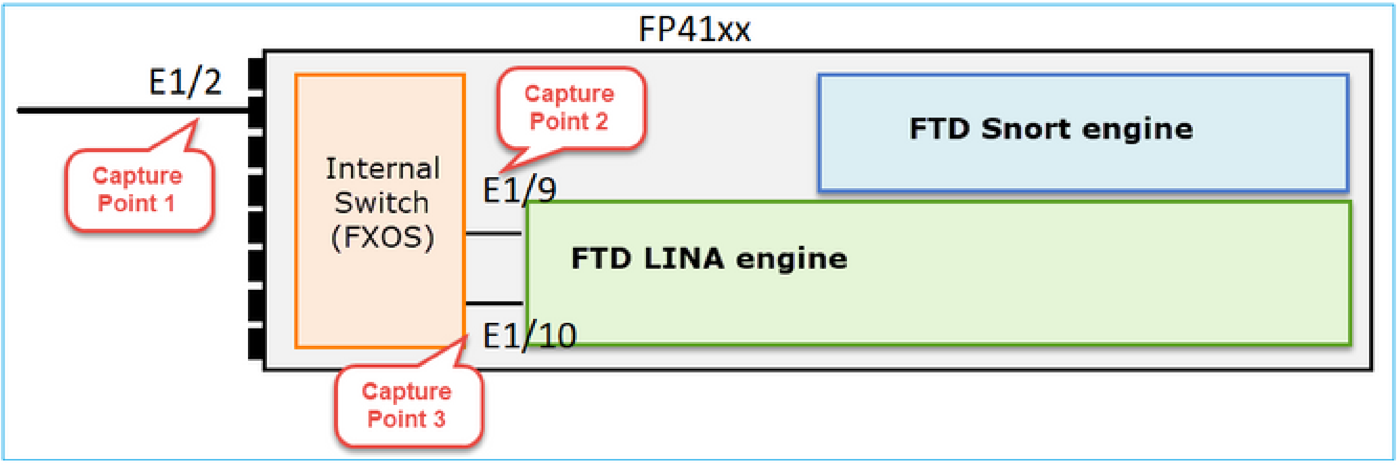
```
firepower#

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
```

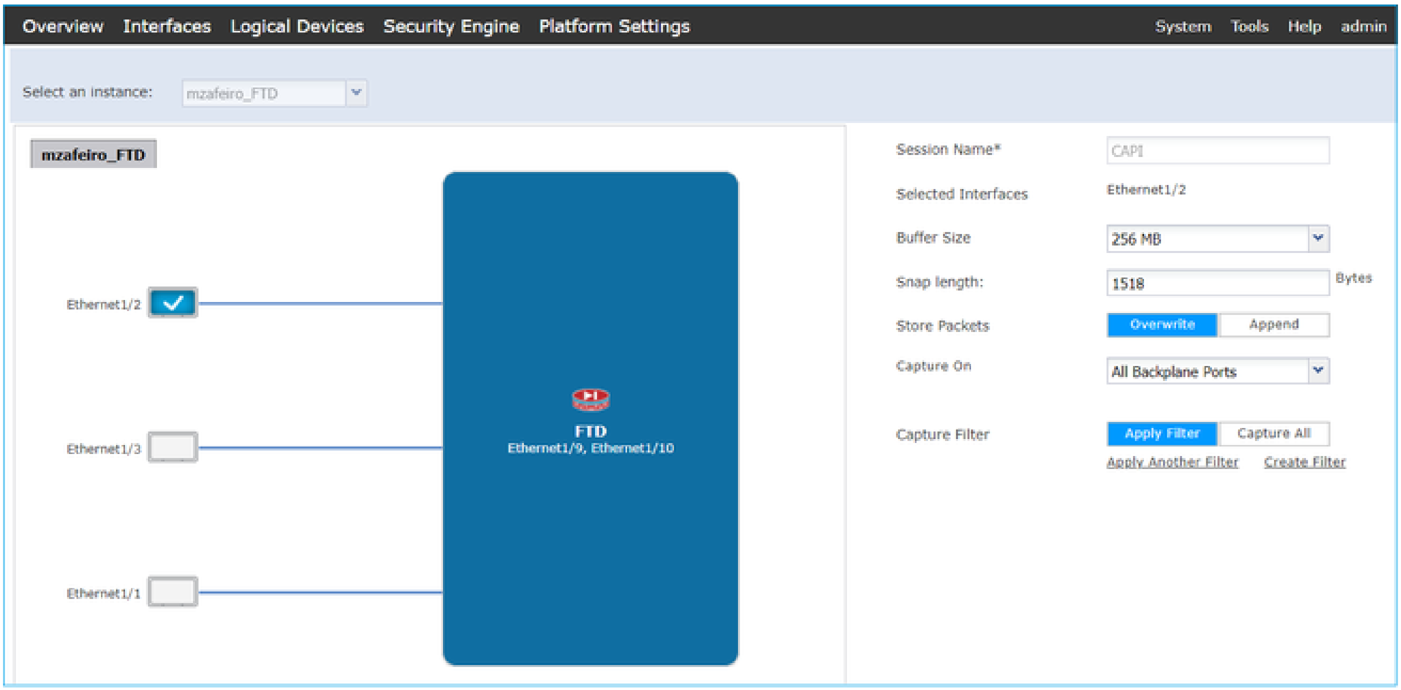
重點：

1. 輸入和輸出介面相同(U-turn)。
2. 位元組數具有非常大的值(~5 GB)。
3. 標誌「o」表示流量分流（硬體加速流量）。這就是為什麼FTD擷取不顯示任何封包。僅41xx和93xx平台支援流量分流。在本例中，裝置是41xx。

行動2.獲取機箱級捕獲。

連線到Firepower機箱管理器，並在入口介面（本例中為E1/2）和背板介面（E1/9和E1/10）上啟用捕獲，如下圖所示：





Select an instance: mzafeiro_FTD

Session Name* CAPI

Selected Interfaces Ethernet1/2

Buffer Size 256 MB

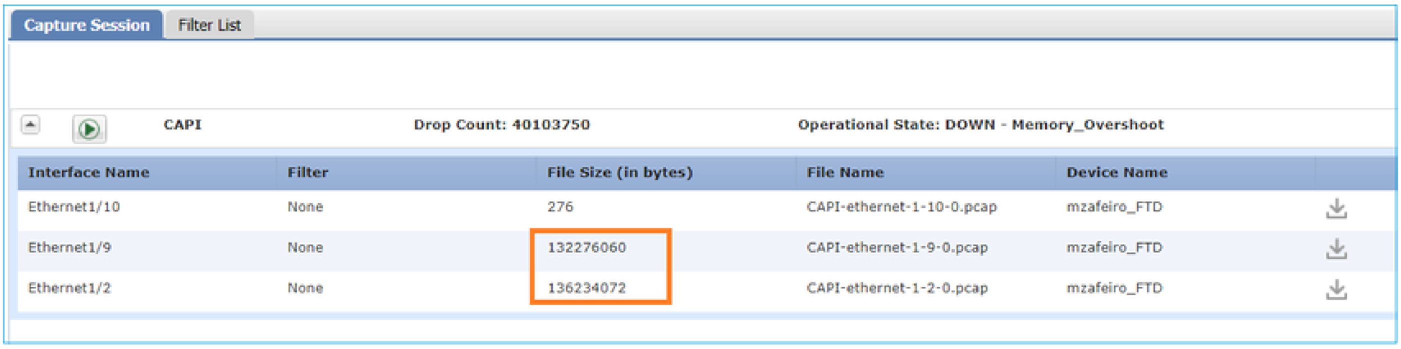
Snap length: 1518 Bytes

Store Packets [✓ Overwrite] Append

Capture On All Backplane Ports

Capture Filter [✓ Apply Filter] Capture All

幾秒鐘後：

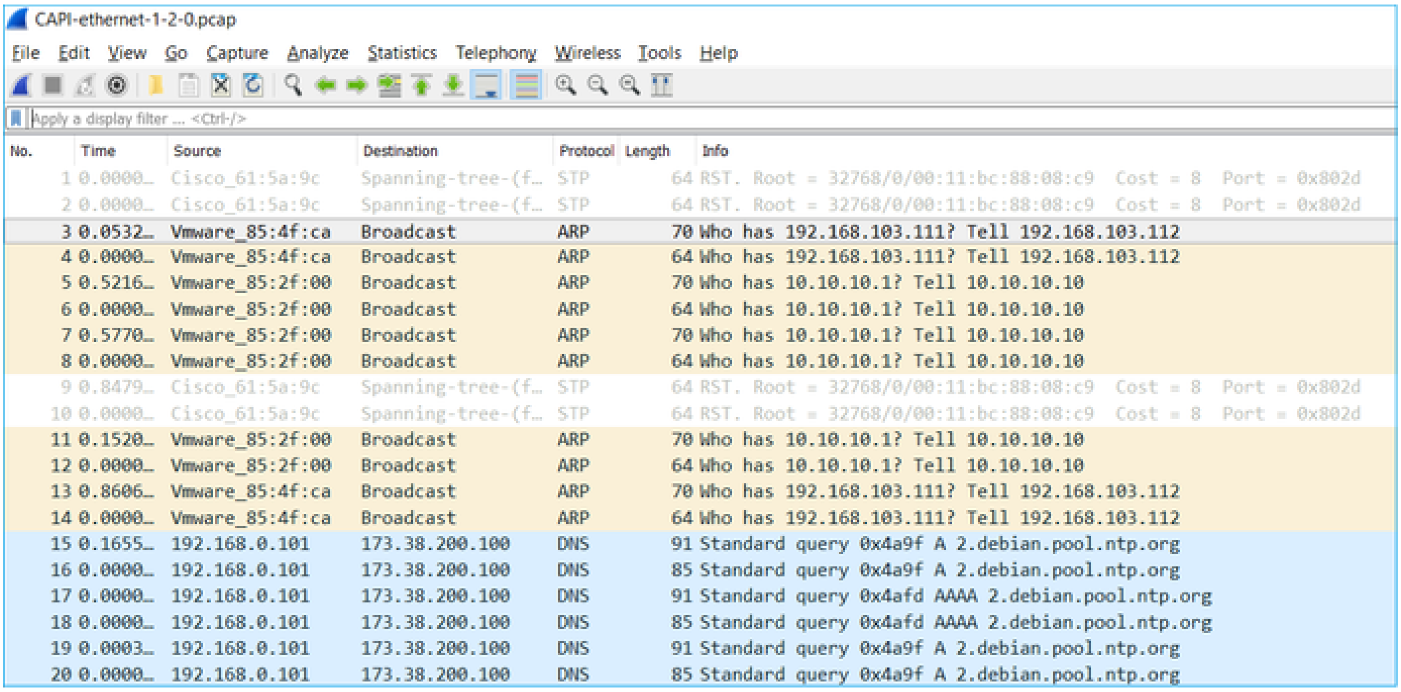


CAPI Drop Count: 40103750 Operational State: DOWN - Memory_Overshoot

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

提示：在Wireshark中，排除VN標籤的資料包，以消除物理介面級別的資料包重複

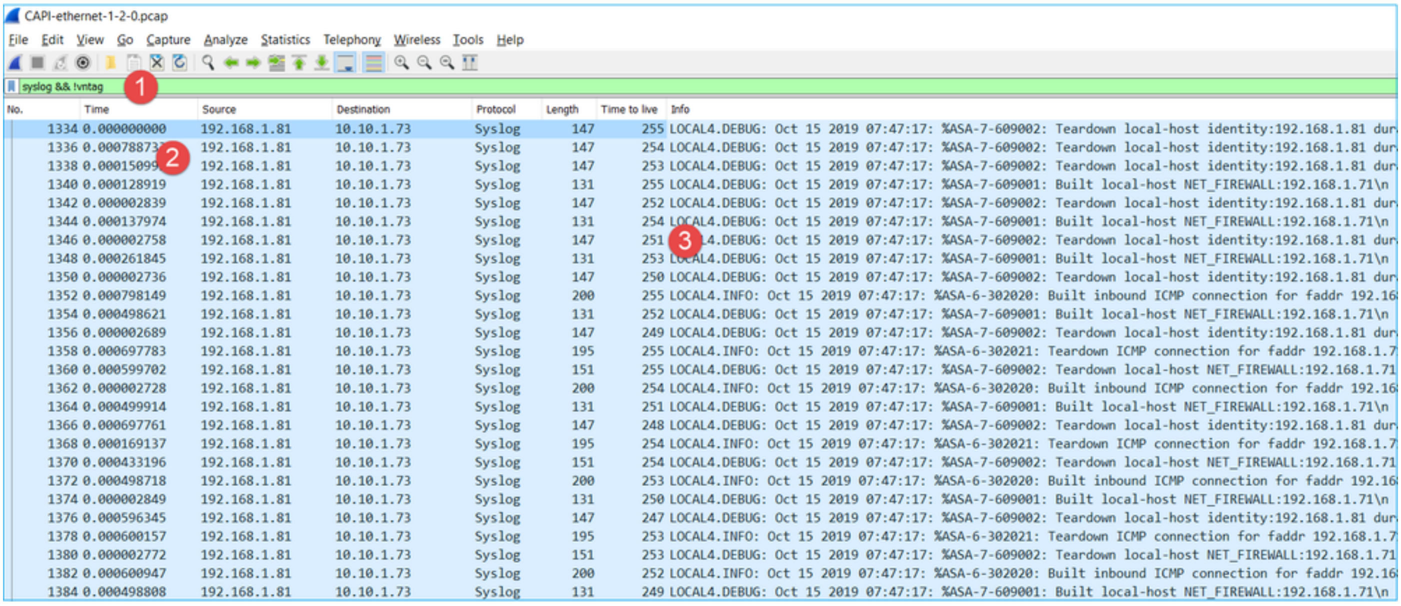
之前：



CAPI-ethernet-1-2-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

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

之後：

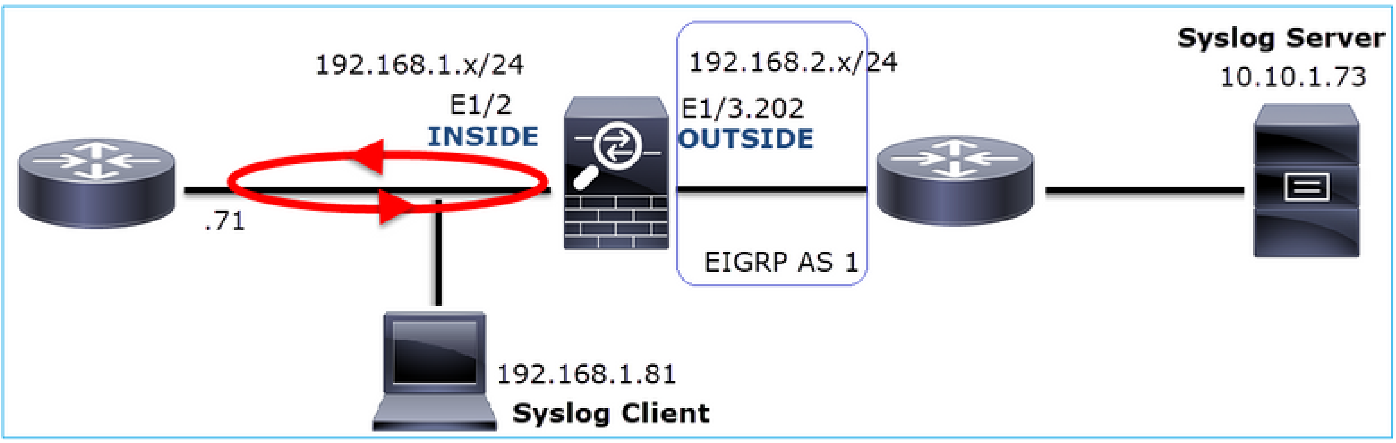


CAPI-ethernet-1-2-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
syslog && !vntag
1
2
3

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

重點：

1. 應用顯示過濾器可刪除資料包重複項並僅顯示系統日誌。
2. 資料包之間的差異處於微秒級別。這表示封包速率非常高。
3. 生存時間(TTL)值持續減小。這表示封包回圈。



行動3.使用Packet Tracer。

由於封包沒有經過防火牆LINA引擎，因此您無法執行即時追蹤（擷取/追蹤），但可以使用packet Tracer追蹤模擬封包：

<#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
```

行動4.確認FTD路由。

檢查防火牆路由表，檢視是否存在路由問題：

<#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
```

重點：

1. 該路由指向正確的出口介面。
2. 路由幾分鐘前獲知(0:02:37)。

行動5.確認連線正常運行時間。

檢查連線正常運行時間，檢視建立此連線的時間：

<#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
```

要點：

1. 連線是在約4分鐘前建立的（這是在路由表中安裝EIGRP路由之前）

行動6.清除已建立的連線。

在這種情況下，資料包匹配已建立的連線，並被路由到錯誤的輸出介面；這會導致環路。這是因為防火牆的操作順序：

1. 已建立的連線查詢（其優先順序高於全域性路由表查詢）。
2. 網路地址轉換(NAT)查詢 — UN-NAT（目標NAT）階段優先於PBR和路由查詢。
3. 原則型路由(PBR)
4. 全域性路由表查詢

由於連線從不超時（Syslog客戶端持續傳送資料包，而UDP連線空閒超時為2分鐘），因此需要手動清除連線：

<#root>

```
firepower#

clear conn address 10.10.1.73 address 192.168.1.81 protocol udp port 514

1 connection(s) deleted.
```

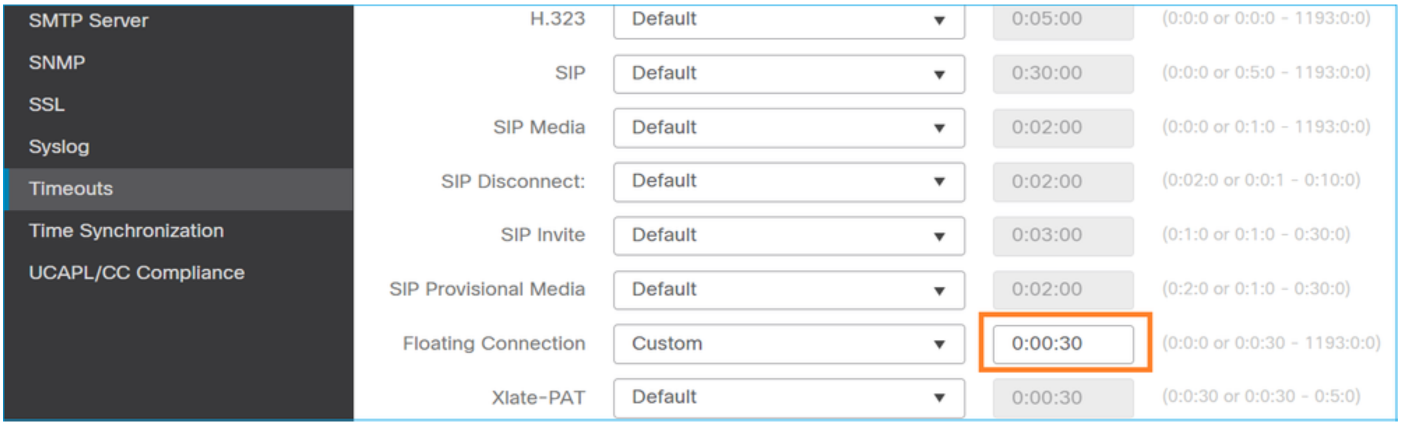
驗證是否已建立新連線：

<#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
```

行動7.配置浮動連線超時。

這是解決此問題並避免次優路由的正確解決方案，對於UDP資料流尤其如此。導覽至Devices > Platform Settings > Timeouts，然後設定值：



SMTP Server
SNMP
SSL
Syslog
Timeouts
Time Synchronization
UCAPL/CC Compliance

H.323	Default	0:05:00	(0:0:0 or 0:0:0 - 1193:0:0)
SIP	Default	0:30:00	(0:0:0 or 0:5:0 - 1193:0:0)
SIP Media	Default	0:02:00	(0:0:0 or 0:1:0 - 1193:0:0)
SIP Disconnect:	Default	0:02:00	(0:02:0 or 0:0:1 - 0:10:0)
SIP Invite	Default	0:03:00	(0:1:0 or 0:1:0 - 0:30:0)
SIP Provisional Media	Default	0:02:00	(0:2:0 or 0:1:0 - 0:30:0)
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)

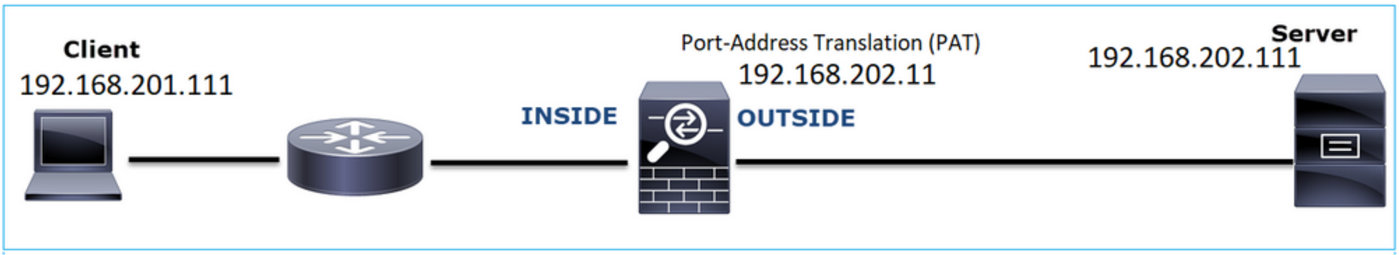
有關浮動連線埠逾時的詳細資訊，請參閱命令參考：

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

案例9.HTTPS連線問題（場景1）

問題描述：無法建立客戶端192.168.201.105和伺服器192.168.202.101之間的HTTPS通訊

下圖顯示拓撲：



受影響的流：

源IP:192.168.201.111

Dst IP:192.168.202.111

協定：TCP 443(HTTPS)

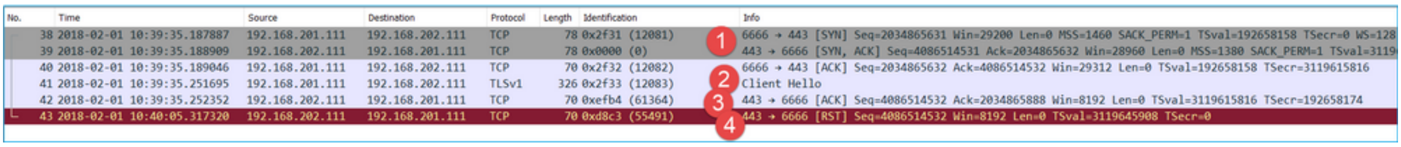
捕獲分析

在FTD LINA引擎上啟用擷取：

由於埠地址轉換配置，OUTSIDE捕獲中使用的IP不同。

```
<#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
```

此圖顯示NGFW INSIDE介面上的擷取：

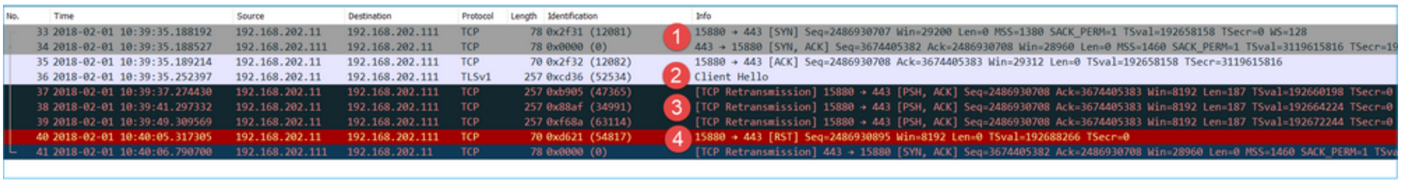


```
No. Time                         Source           Destination      Protocol Length Identification  Info
 38 2018-02-01 10:39:35.187887  192.168.201.111  192.168.202.111  TCP          78 0x2f31 (12081)  6666 → 443 [SYN] Seq=2034865631 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=192658158 TSecr=0 WS=128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

重點：

1. 有一個TCP三次握手。
2. SSL協商啟動。客戶端傳送客戶端Hello消息。
3. 系統向客戶端傳送了TCP ACK。
4. 有一條TCP RST傳送到客戶端。

此圖顯示在NGFW OUTSIDE介面上捕獲的流量。



```
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=19
 35 2018-02-01 10:39:35.189214  192.168.202.11   192.168.202.111  TCP          70 0x2f32 (12082)  15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Win=29312 Len=0 TSval=192658158 TSecr=3119615816
 36 2018-02-01 10:39:35.252397  192.168.202.11   192.168.202.111  TLSv1       257 0xcd36 (52534)  Client Hello
 37 2018-02-01 10:39:37.274430  192.168.202.11   192.168.202.111  TCP         257 0xb905 (47365)  [TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192660198 TSecr=0
 38 2018-02-01 10:39:41.297332  192.168.202.11   192.168.202.111  TCP         257 0x88af (34991)  [TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192664224 TSecr=0
 39 2018-02-01 10:39:49.309569  192.168.202.11   192.168.202.111  TCP         257 0xf68a (63114)  [TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192672244 TSecr=0
 40 2018-02-01 10:40:05.317305  192.168.202.11   192.168.202.111  TCP          70 0xd621 (54817)  15880 → 443 [RST] Seq=2486930895 Win=8192 Len=0 TSval=192688266 TSecr=0
 41 2018-02-01 10:40:06.790700  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 TSva
```

重點：

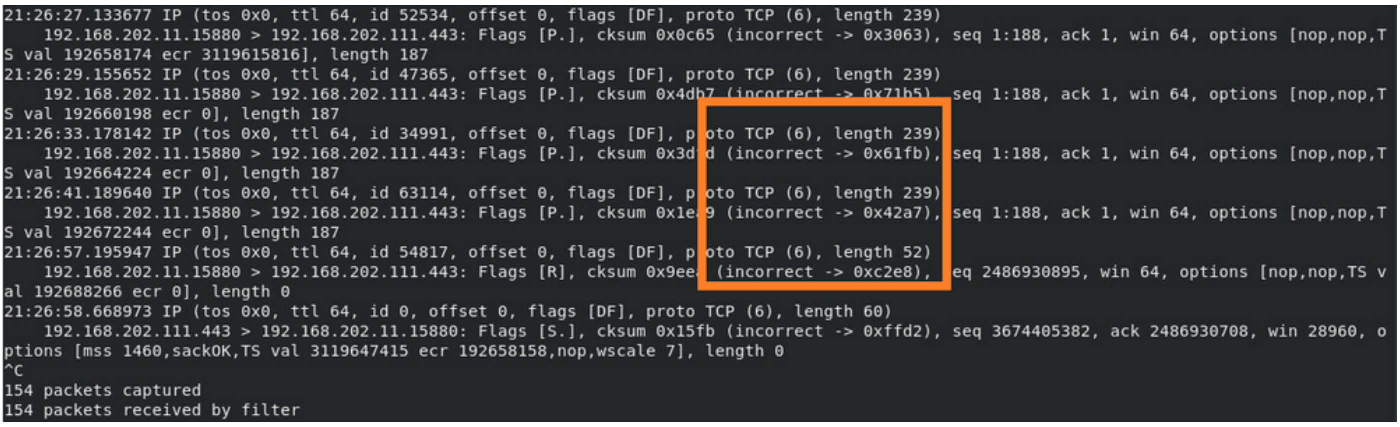
1. 有一個TCP三次握手。
2. SSL協商啟動。客戶端傳送客戶端Hello消息。
3. 存在從防火牆向伺服器傳送的TCP重新傳輸。
4. 有一條TCP RST傳送到伺服器。

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.獲取其他捕獲。

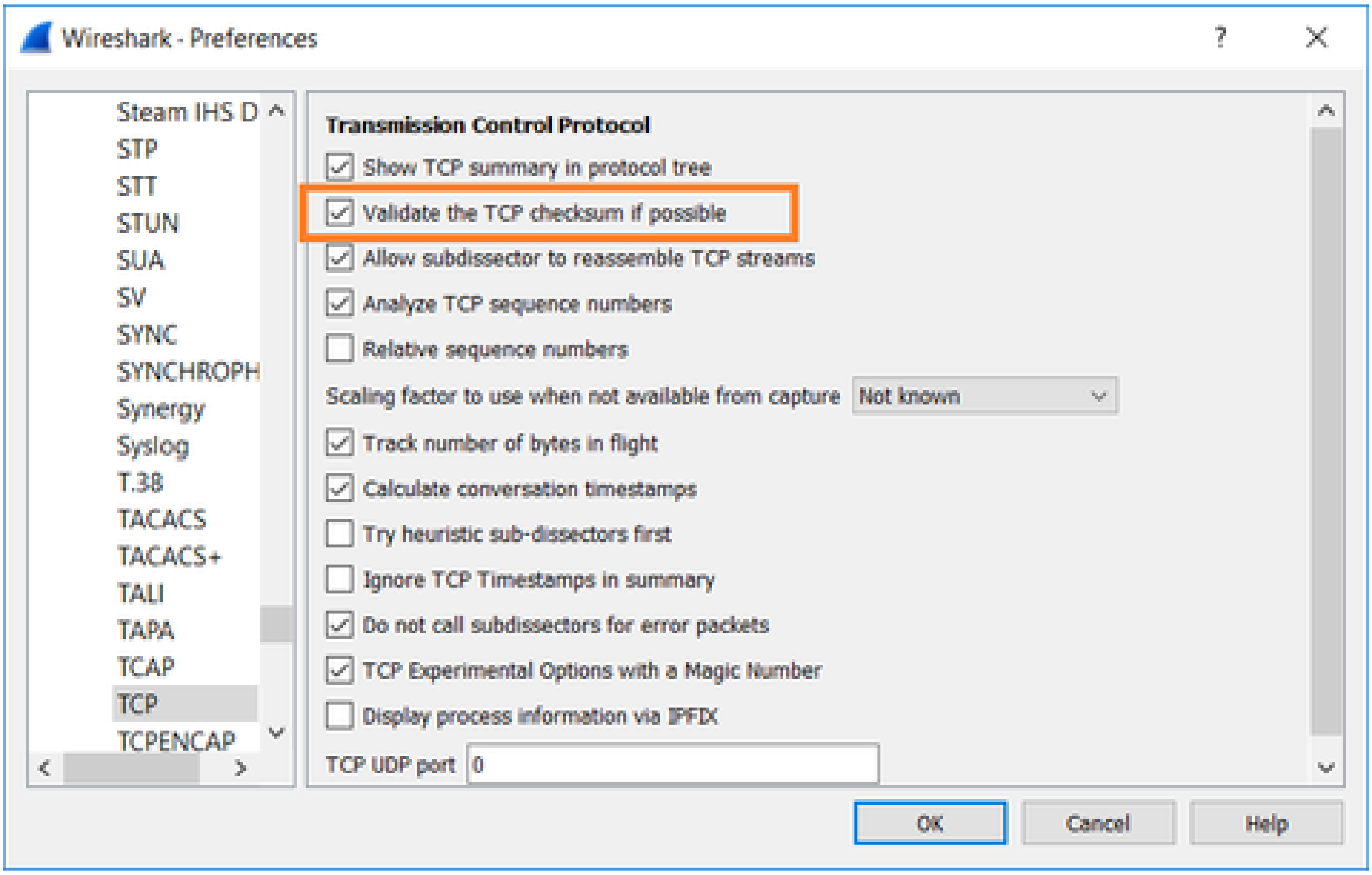
在伺服器上截獲的資訊顯示，伺服器收到的TLS客戶端Hello的TCP校驗和已損壞，然後以靜默方式丟棄它們（沒有指向客戶端的TCP RST或任何其他回複資料包）：



```
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], p oto TCP (6), length 239)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x3d d (incorrect -> 0x61fb), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192664224 ecr 0], length 187
21:26:41.189640 IP (tos 0x0, ttl 64, id 63114, offset 0, flags [DF], p oto TCP (6), length 239)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x1e 9 (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], p oto TCP (6), length 52)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [R], cksum 0x9ee  (incorrect -> 0xc2e8),  eq 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
```

當你把所有東西放在一起時：

在這種情況下，為了理解，需要在Wireshark上啟用Validate the TCP checksum if possible選項。導覽至Edit > Preferences > Protocols > TCP，如下圖所示。



Transmission Control Protocol

☑ Show TCP summary in protocol tree

☑ Validate the TCP checksum if possible

☑ Allow subdissector to reassemble TCP streams

☑ Analyze TCP sequence numbers

□ Relative sequence numbers

Scaling factor to use when not available from capture: Not known

☑ Track number of bytes in flight

☑ Calculate conversation timestamps

□ Try heuristic sub-dissectors first

□ Ignore TCP Timestamps in summary

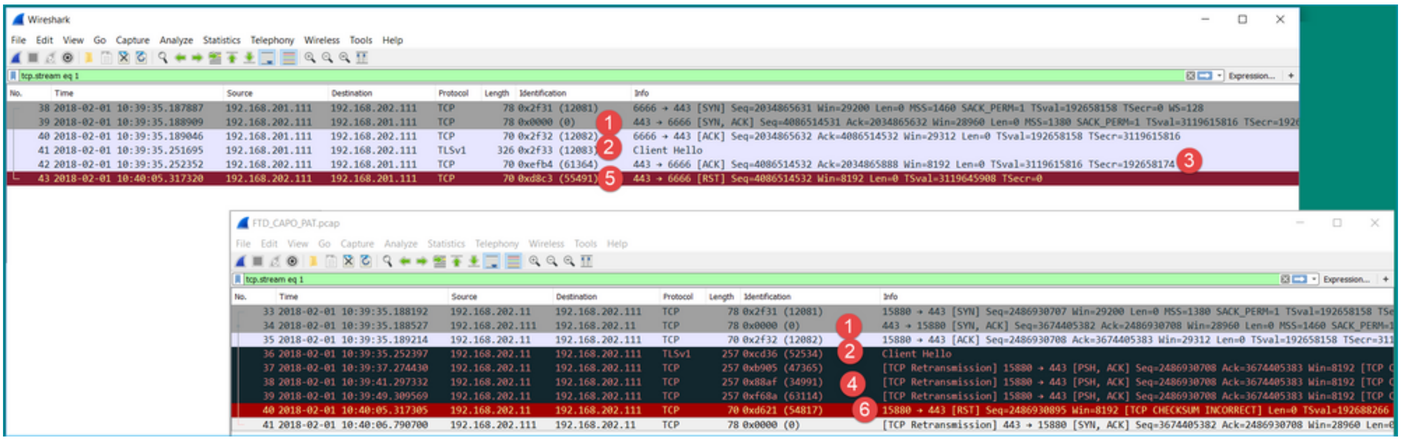
☑ Do not call subdissectors for error packets

☑ TCP Experimental Options with a Magic Number

□ Display process information via IPFIX

TCP UDP port: 0

在這種情況下，將擷取並列放置以取得完整畫面是很有用的：



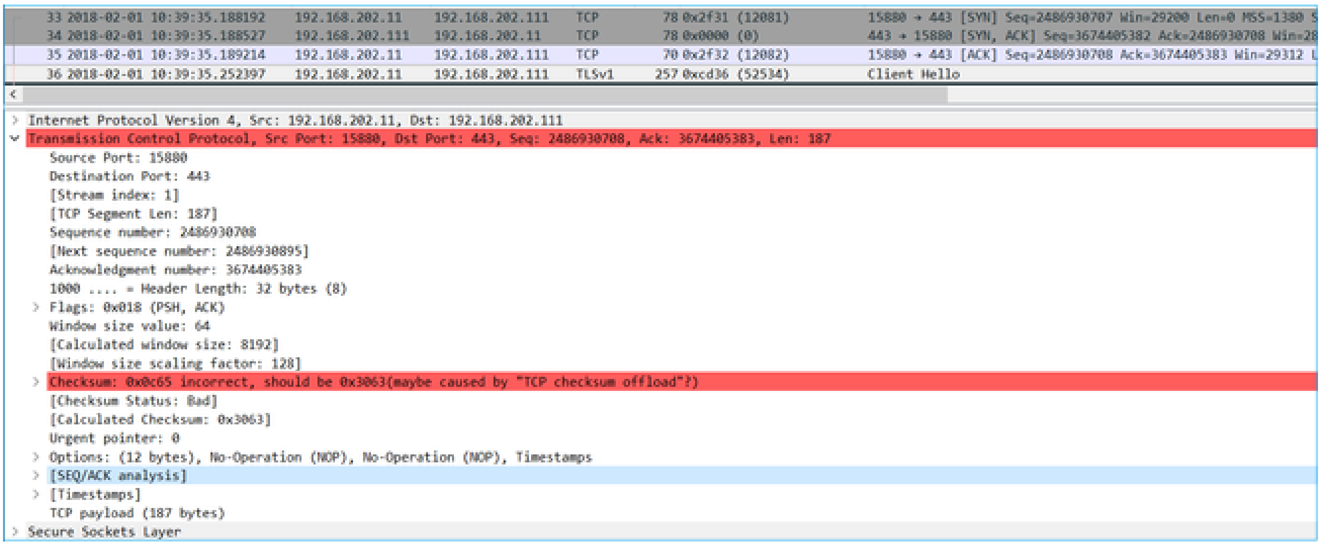
Wireshark
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
1
2
3
5
FTD_CAPO_PAT.pcap
1
2
4
6

```
tcp.stream eq 1
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=192
  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
```

```
tcp.stream eq 1
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 TSe
  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
  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=311
  36 2018-02-01 10:39:35.252397  192.168.202.11   192.168.202.111  TLSv1    257 0xcd36 (52534)  Client Hello
  37 2018-02-01 10:39:37.274430  192.168.202.11   192.168.202.111  TCP      257 0xb905 (47365)  [TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 [TCP C
  38 2018-02-01 10:39:41.297332  192.168.202.11   192.168.202.111  TCP      257 0x88af (34991)  [TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 [TCP C
  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 [TCP C
  40 2018-02-01 10:40:05.317305  192.168.202.11   192.168.202.111  TCP       70 0xd621 (54817)  15880 → 443 [RST] Seq=2486930895 Win=8192 [TCP CHECKSUM INCORRECT] Len=0 TSval=192688266
  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
```

重點：

1. 有一個TCP三次握手。IP ID相同。這表示流量不是由防火牆代理的。
2. TLS客戶端Hello來自具有IP ID 12083的客戶端。資料包由防火牆代理（本例中防火牆配置了TLS解密策略），並且IP ID更改為52534。此外，封包TCP總和檢查碼已損毀（由於軟體缺陷，該缺陷稍後被修復）。
3. 防火牆處於TCP代理模式並向客戶端傳送ACK（偽裝伺服器）。



```
  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 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 Win=28
  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 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
v 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
```

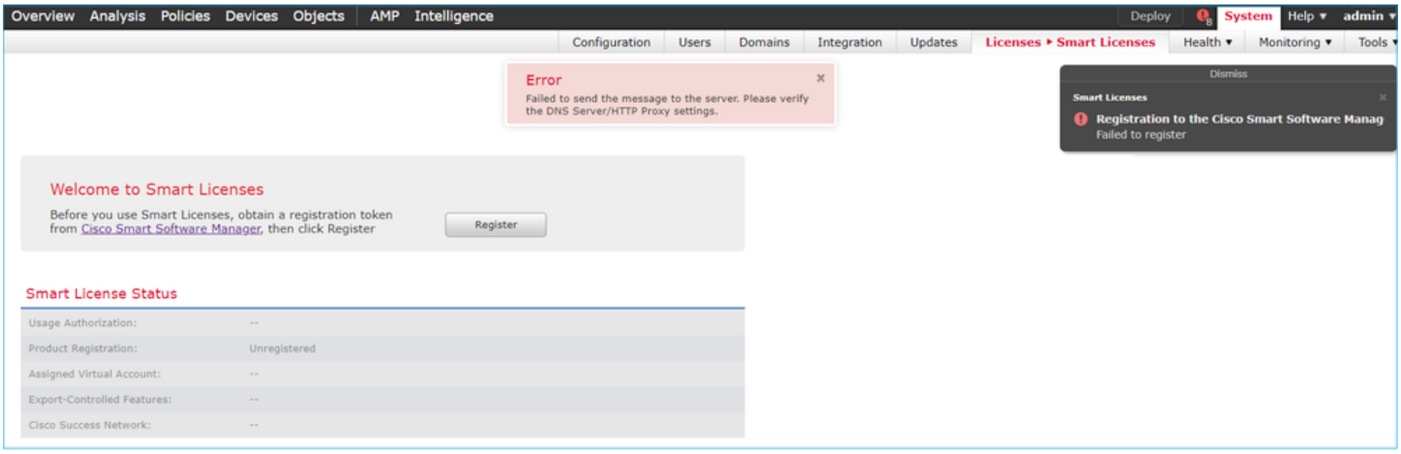
4. 防火牆不會收到來自伺服器的任何TCP ACK資料包，而是重新傳輸TLS客戶端Hello消息。這也是由於防火牆已啟用TCP代理模式所致。
5. 約30秒後，防火牆會放棄並向使用者端傳送TCP RST。
6. 防火牆向伺服器傳送TCP RST。

供參考：

Firepower TLS/SSL握手處理

案例10.HTTPS連線問題（場景2）

問題描述：FMC智慧許可證註冊失敗。



Error

Failed to send the message to the server. Please verify the DNS Server/HTTP Proxy settings.

Smart Licenses

Registration to the Cisco Smart Software Manag

Failed to register

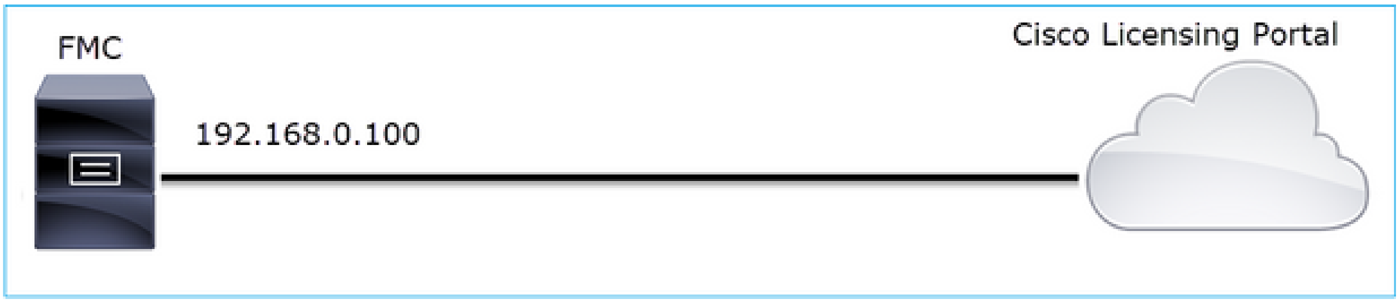
Welcome to Smart Licenses

Before you use Smart Licenses, obtain a registration token from Cisco Smart Software Manager, then click Register

Smart License Status

Usage Authorization:	--
Product Registration:	Unregistered
Assigned Virtual Account:	--
Export-Controlled Features:	--
Cisco Success Network:	--

下圖顯示拓撲：



受影響的流：

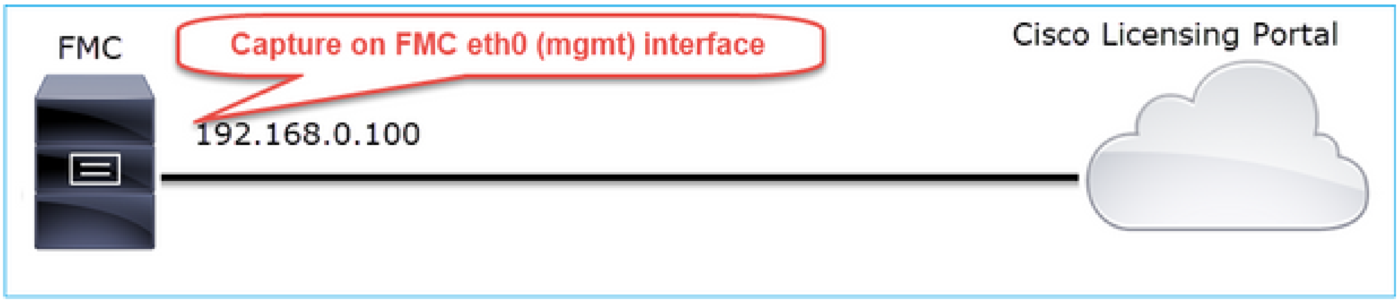
源IP:192.168.0.100

Dst:tools.cisco.com

協定：TCP 443(HTTPS)

捕獲分析

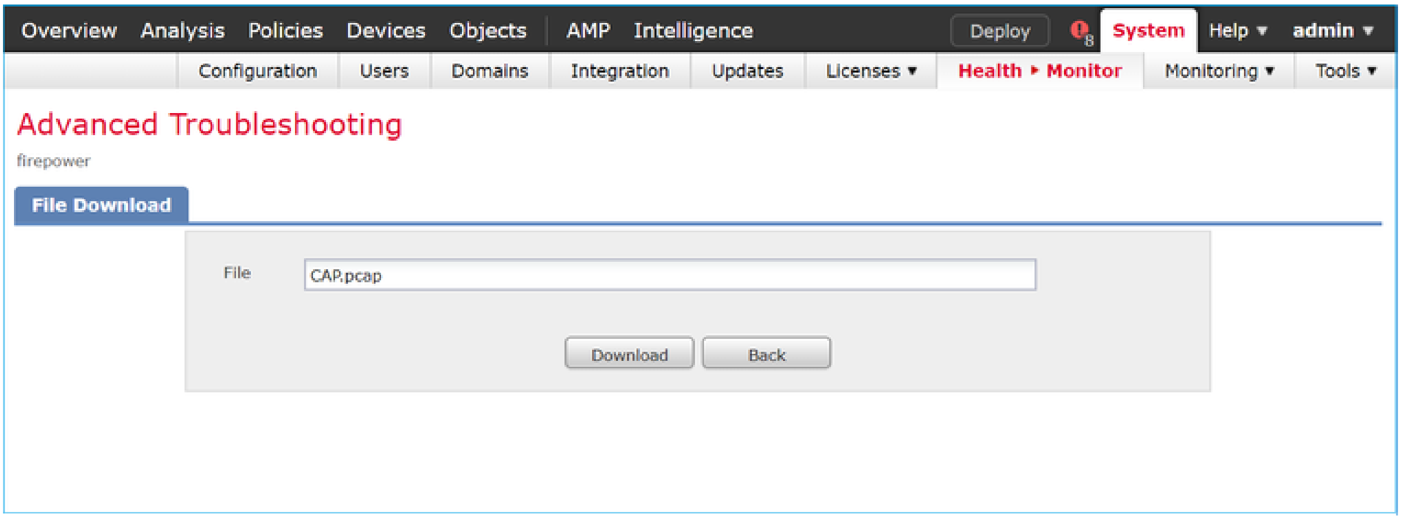
在FMC管理介面上啟用捕獲：



再次嘗試註冊。出現錯誤資訊後，按CTRL-C停止捕獲：

```
<#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#
```

從FMC中收集捕獲(System > Health > Monitor，選擇裝置並選擇Advanced Troubleshooting)，如下圖所示：



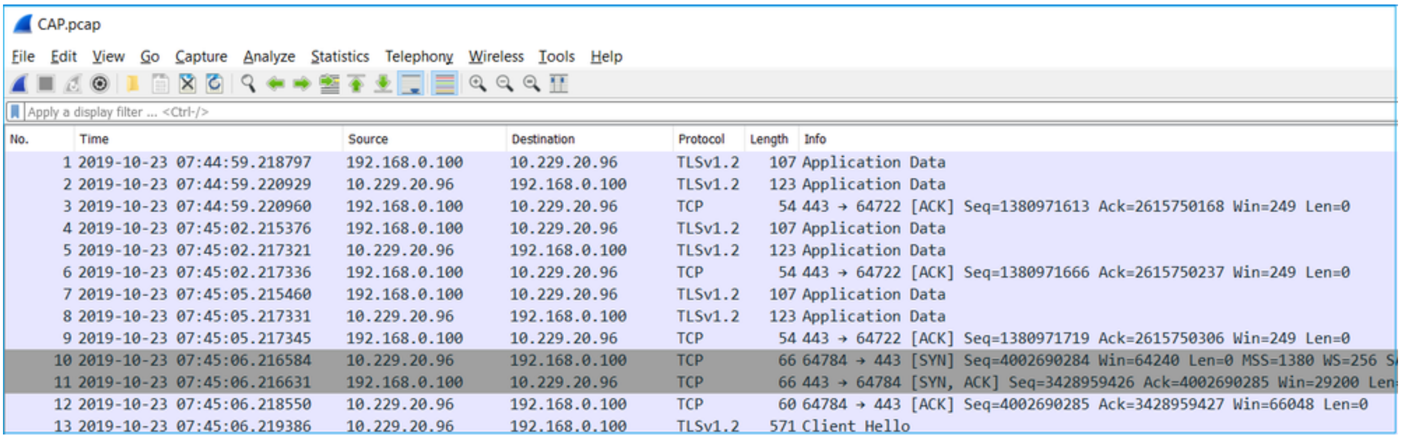
Advanced Troubleshooting

firepower

File Download

File: CAP.pcap

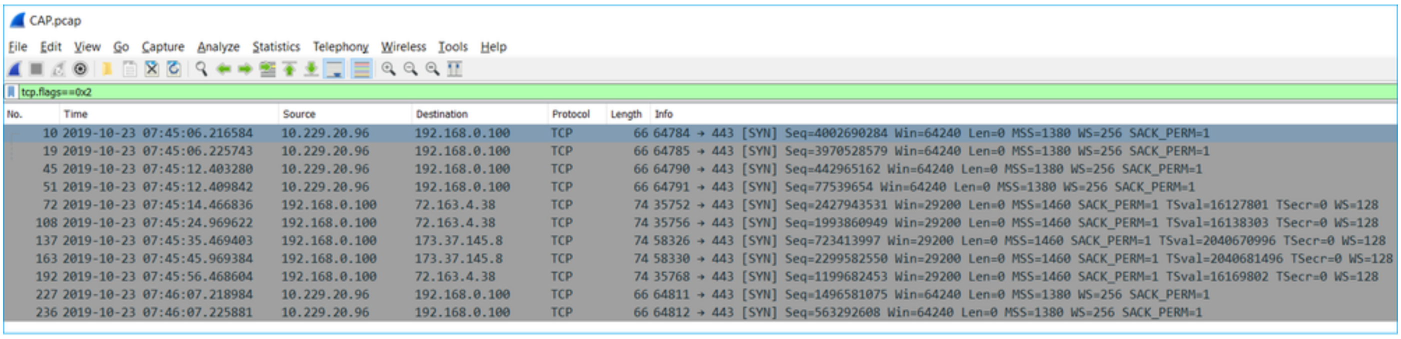
該圖顯示Wireshark上的FMC捕獲：


CAP.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

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

提示：若要檢查是否已捕獲所有新TCP會話，請在Wireshark上使用tcp.flags==0x2顯示過濾器

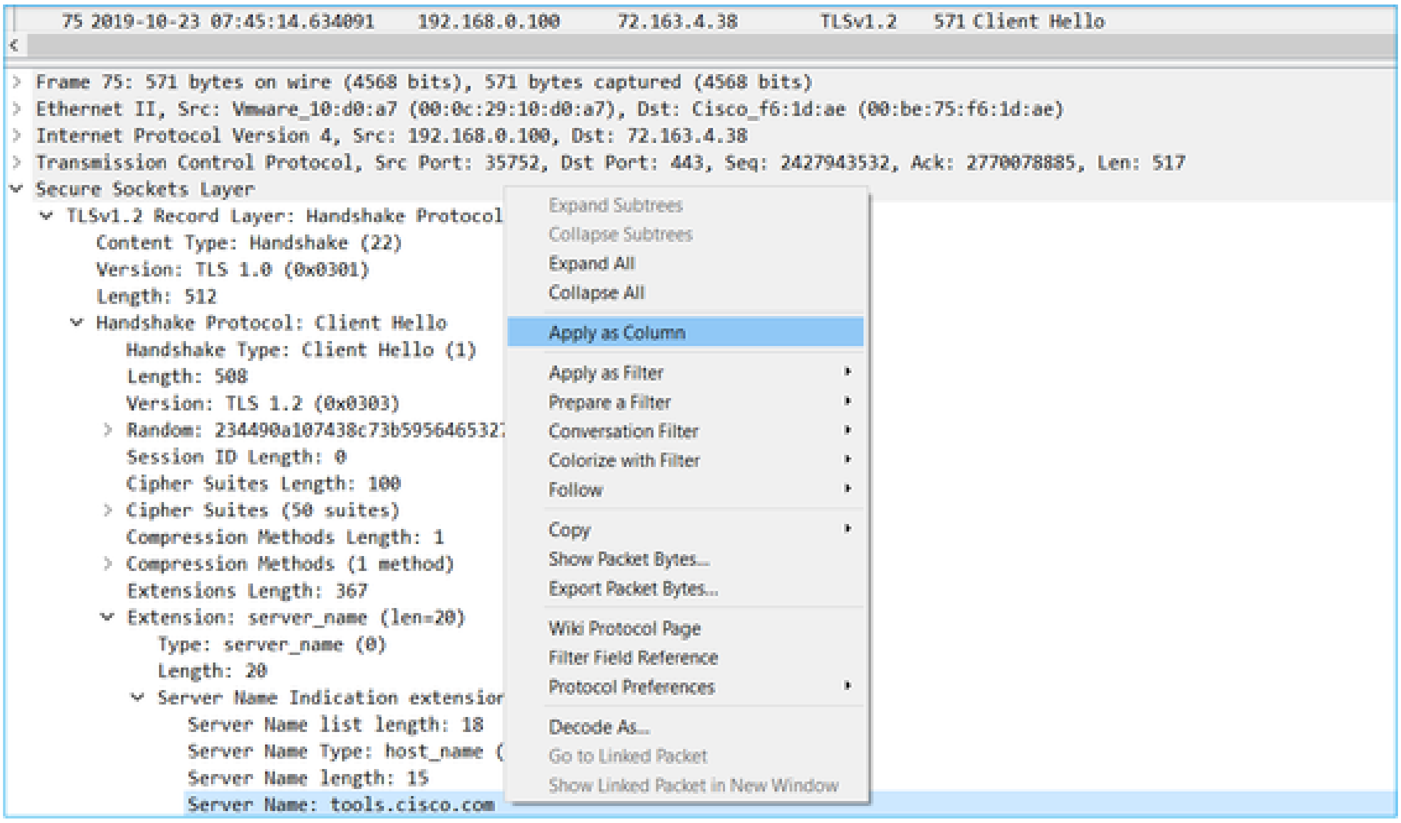
。這會過濾所有擷取的TCP SYN封包。


CAP.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags==0x2

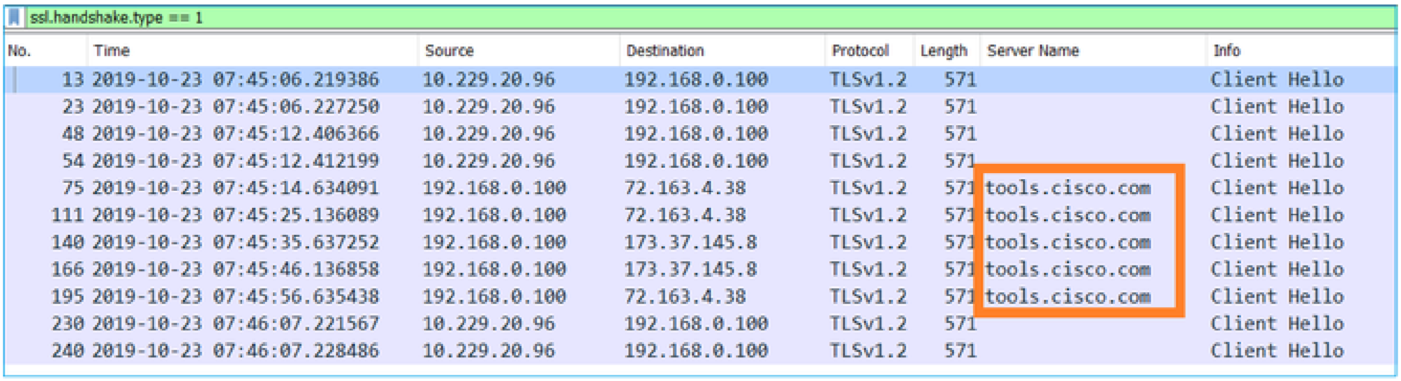
```
No.  Time                        Source          Destination     Protocol  Length Info
  10 2019-10-23 07:45:06.216584  10.229.20.96    192.168.0.100   TCP       66 64784 → 443 [SYN] Seq=4002690284 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1
  19 2019-10-23 07:45:06.225743  10.229.20.96    192.168.0.100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

提示：從SSL客戶端Hello中應用伺服器名稱作為列。



```
    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
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Handshake Protocol
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 508
        Version: TLS 1.2 (0x0303)
      > Random: 234490a107438c73b5956465327
        Session ID Length: 0
        Cipher Suites Length: 100
      > Cipher Suites (50 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 367
      v Extension: server_name (len=20)
          Type: server_name (0)
          Length: 20
        v Server Name Indication extension
            Server Name list length: 18
            Server Name Type: host_name (
            Server Name length: 15
            Server Name: tools.cisco.com
```

提示：應用此顯示過濾器以僅檢視Client Hello消息ssl.handshake.type == 1

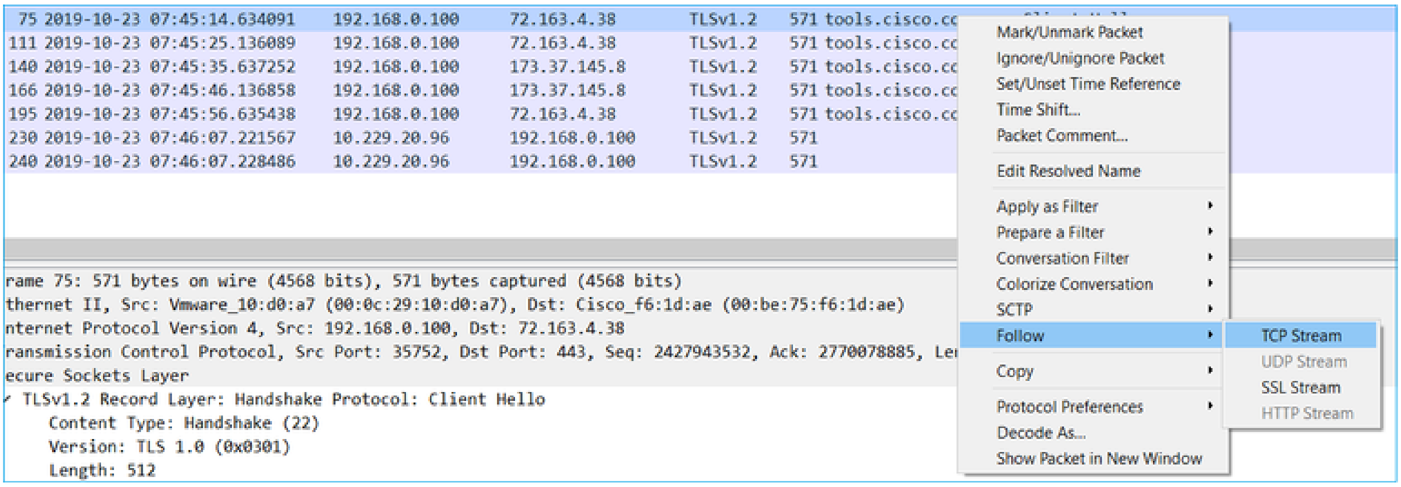


ssl.handshake.type == 1

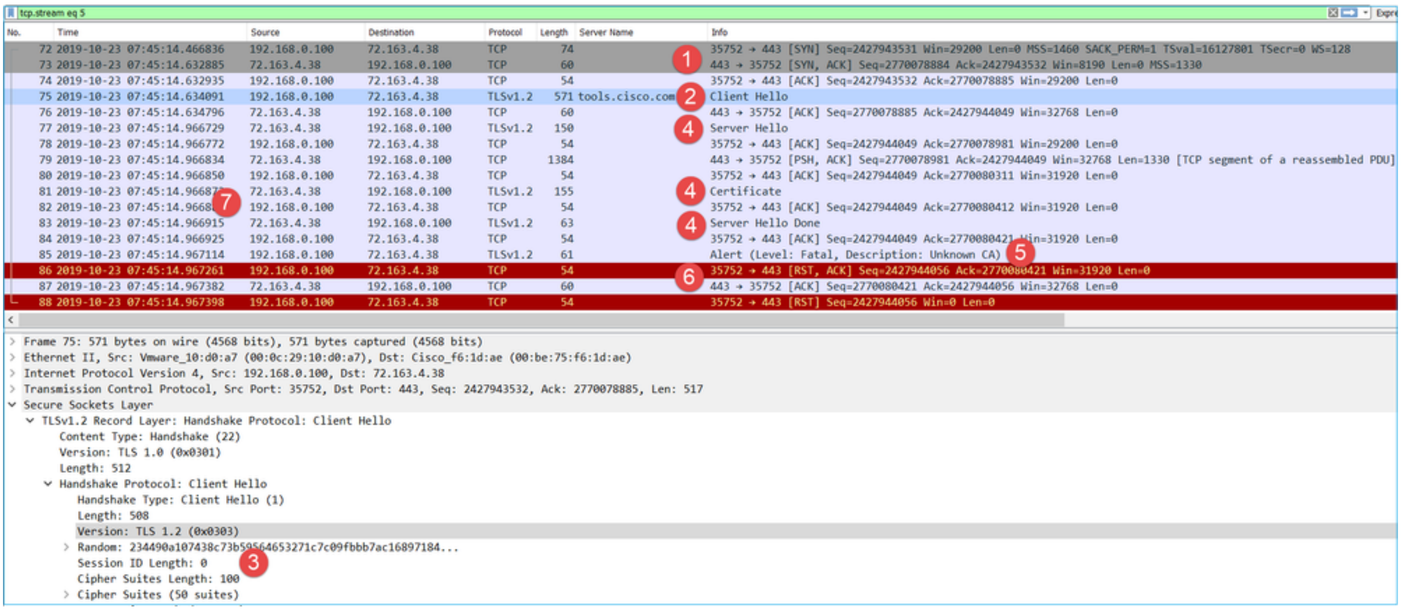
```
No.  Time                        Source          Destination     Protocol  Length  Server Name      Info
  13 2019-10-23 07:45:06.219386  10.229.20.96    192.168.0.100   TLSv1.2   571                      Client Hello
  23 2019-10-23 07:45:06.227250  10.229.20.96    192.168.0.100   TLSv1.2   571                      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
```

註：撰寫本文時，智慧許可門戶(tools.cisco.com)使用以下IP:72.163.4.38、173.37.145.8

按照其中一個TCP資料流執行(Follow > TCP Stream)，如下圖所示。



```
 75 2019-10-23 07:45:14.634091    192.168.0.100     72.163.4.38      TLSv1.2    571 tools.cisco.cc
111 2019-10-23 07:45:25.136089    192.168.0.100     72.163.4.38      TLSv1.2    571 tools.cisco.cc
140 2019-10-23 07:45:35.637252    192.168.0.100     173.37.145.8     TLSv1.2    571 tools.cisco.cc
166 2019-10-23 07:45:46.136858    192.168.0.100     173.37.145.8     TLSv1.2    571 tools.cisco.cc
195 2019-10-23 07:45:56.635438    192.168.0.100     72.163.4.38      TLSv1.2    571 tools.cisco.cc
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, Le
ecure Sockets Layer
 TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 512
```

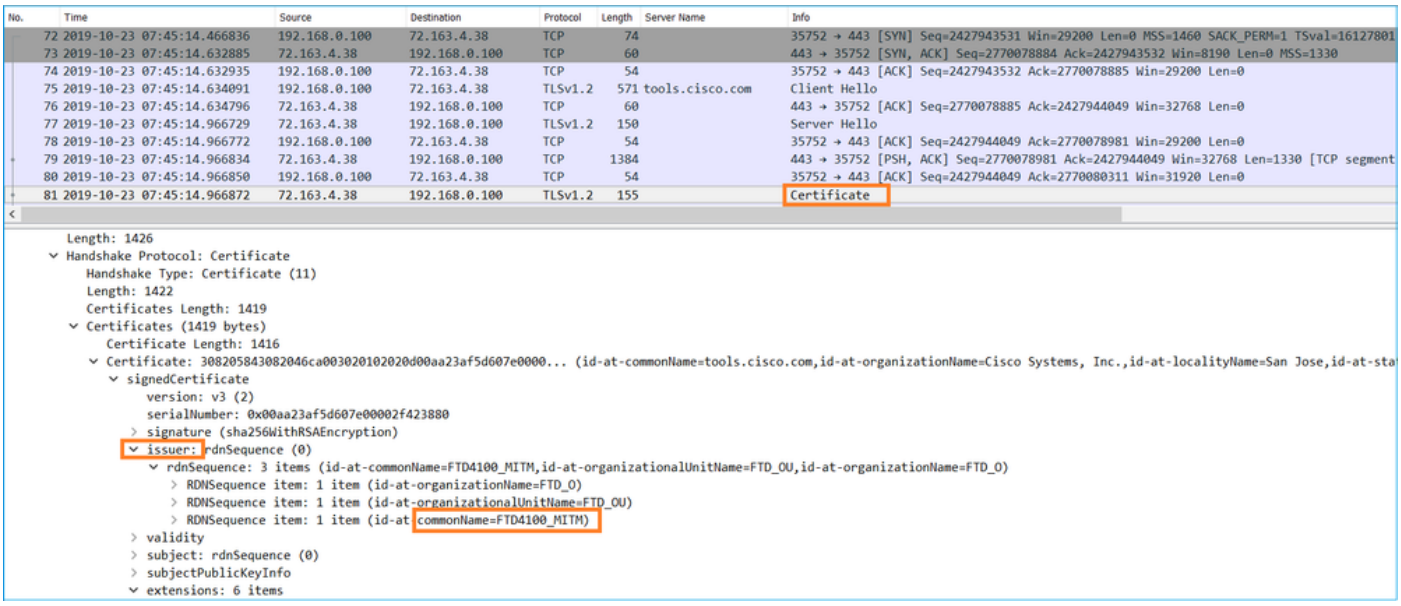


```
tcp.stream eq 5

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       1384                      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 07:45:14.96687   72.163.4.38     192.168.0.100   TLSv1.2    155                      Certificate
 82 2019-10-23 07:45:14.9668    192.168.0.100   72.163.4.38     TCP         54                      35752 → 443 [ACK] Seq=2427944049 Ack=2770080412 Win=31920 Len=0
 83 2019-10-23 07:45:14.966915  72.163.4.38     192.168.0.100   TLSv1.2     63                      Server Hello Done
 84 2019-10-23 07:45:14.966925  192.168.0.100   72.163.4.38     TCP         54                      35752 → 443 [ACK] Seq=2427944049 Ack=2770080421 Win=31920 Len=0
 85 2019-10-23 07:45:14.967114  192.168.0.100   72.163.4.38     TLSv1.2     61                      Alert (Level: Fatal, Description: Unknown CA)
 86 2019-10-23 07:45:14.967261  192.168.0.100   72.163.4.38     TCP         54                      35752 → 443 [RST, ACK] Seq=2427944056 Ack=2770080421 Win=31920 Len=0
 87 2019-10-23 07:45:14.967382  72.163.4.38     192.168.0.100   TCP         60                      443 → 35752 [ACK] Seq=2770080421 Ack=2427944056 Win=32768 Len=0
 88 2019-10-23 07:45:14.967398  192.168.0.100   72.163.4.38     TCP         54                      35752 → 443 [RST] Seq=2427944056 Win=0 Len=0

> Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits)
> 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
v Secure Sockets Layer
  v TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 508
        Version: TLS 1.2 (0x0303)
      > Random: 234490a107438c73b59564653271c7c09fbbb7ac16897184...
        Session ID Length: 0
        Cipher Suites Length: 100
      > Cipher Suites (50 suites)
```

重點：

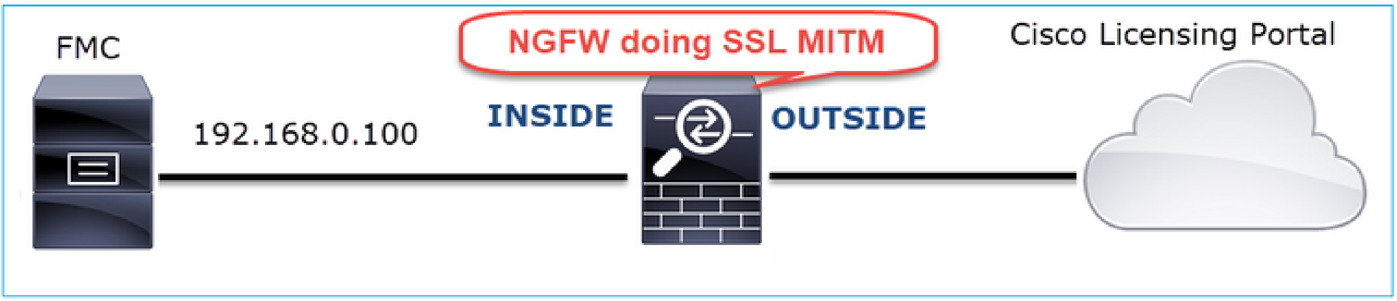
1. 有一個TCP三次握手。
2. 客戶端(FMC)向智慧許可門戶傳送SSL客戶端Hello消息。
3. SSL會話ID為0。這意味著它不是續會。
4. 目標伺服器會使用Server Hello、Certificate和Server Hello Done消息進行回覆。
5. 客戶端傳送有關「未知CA」的SSL致命警報。
6. 使用者端傳送TCP RST以關閉作業階段。
7. 整個TCP會話持續時間（從建立到關閉）約為0.5秒。

選擇Server Certificate，然後展開issuer欄位以檢視commonName。在這種情況下，「公用名」顯示一種執行「中間人」(MITM)的裝置。



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

      Length: 1426
    ˅ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1422
        Certificates Length: 1419
      ˅ Certificates (1419 bytes)
          Certificate Length: 1416
        ˅ Certificate: 308205843082046ca003020102020d00aa23af5d607e0000... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose,id-at-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)
                > RDNSequence item: 1 item (id-at-organizationName=FTD_O)
                > RDNSequence item: 1 item (id-at-organizationalUnitName=FTD_OU)
                > RDNSequence item: 1 item (id-at-commonName=FTD4100_MITM)
            > validity
            > subject: rdnSequence (0)
            > subjectPublicKeyInfo
            ˅ extensions: 6 items
```

如下圖所示：

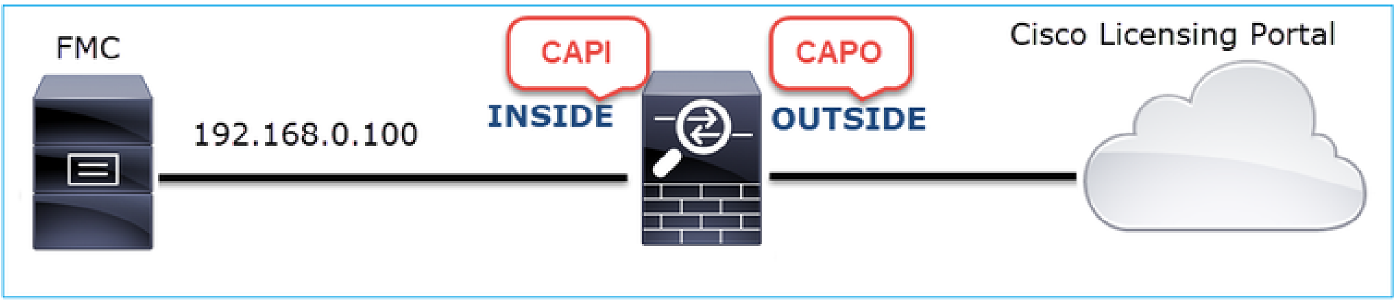


建議的操作

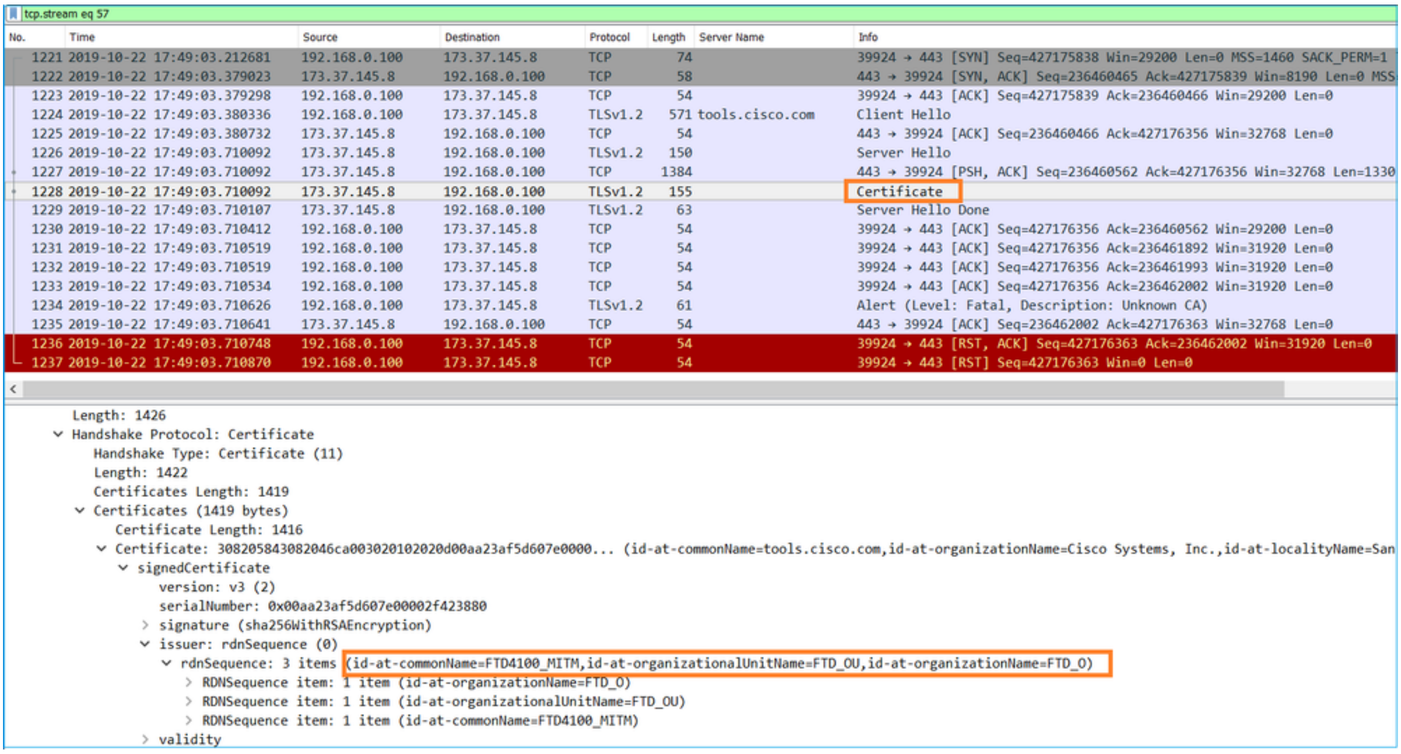
本節所列的行動旨在進一步縮小問題範圍。

操作1.獲取其他捕獲。

在傳輸防火牆裝置上捕獲以下內容：

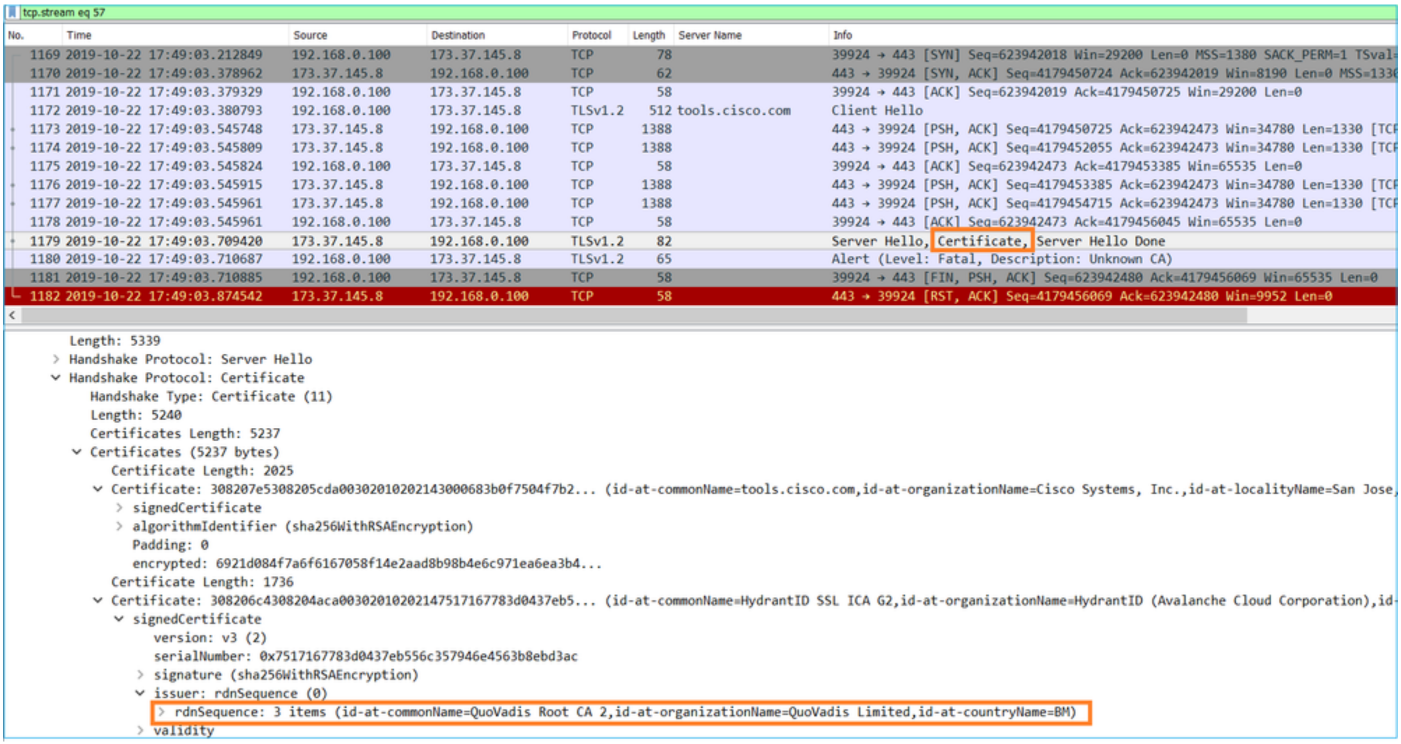


CAPI顯示：



```
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
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)
                › RDNSequence item: 1 item (id-at-organizationName=FTD_O)
                › RDNSequence item: 1 item (id-at-organizationalUnitName=FTD_OU)
                › RDNSequence item: 1 item (id-at-commonName=FTD4100_MITM)
            › validity
```

CAPO顯示：



```
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=1330
1171 2019-10-22 17:49:03.379329 192.168.0.100  173.37.145.8   TCP        58                   39924 → 443 [ACK] Seq=623942019 Ack=4179450725 Win=29200 Len=0
1172 2019-10-22 17:49:03.380793 192.168.0.100  173.37.145.8   TLSv1.2   512 tools.cisco.com   Client Hello
1173 2019-10-22 17:49:03.545748 173.37.145.8   192.168.0.100  TCP      1388                   443 → 39924 [PSH, ACK] Seq=4179450725 Ack=623942473 Win=34780 Len=1330 [TCP
1174 2019-10-22 17:49:03.545809 173.37.145.8   192.168.0.100  TCP      1388                   443 → 39924 [PSH, ACK] Seq=4179452055 Ack=623942473 Win=34780 Len=1330 [TCP
1175 2019-10-22 17:49:03.545824 192.168.0.100  173.37.145.8   TCP        58                   39924 → 443 [ACK] Seq=623942473 Ack=4179453385 Win=65535 Len=0
1176 2019-10-22 17:49:03.545915 173.37.145.8   192.168.0.100  TCP      1388                   443 → 39924 [PSH, ACK] Seq=4179453385 Ack=623942473 Win=34780 Len=1330 [TCP
1177 2019-10-22 17:49:03.545961 173.37.145.8   192.168.0.100  TCP      1388                   443 → 39924 [PSH, ACK] Seq=4179454715 Ack=623942473 Win=34780 Len=1330 [TCP
1178 2019-10-22 17:49:03.545961 192.168.0.100  173.37.145.8   TCP        58                   39924 → 443 [ACK] Seq=623942473 Ack=4179456045 Win=65535 Len=0
1179 2019-10-22 17:49:03.709420 173.37.145.8   192.168.0.100  TLSv1.2    82                   Server Hello, Certificate, Server Hello Done
1180 2019-10-22 17:49:03.710687 192.168.0.100  173.37.145.8   TLSv1.2    65                   Alert (Level: Fatal, Description: Unknown CA)
1181 2019-10-22 17:49:03.710885 192.168.0.100  173.37.145.8   TCP        58                   39924 → 443 [FIN, PSH, ACK] Seq=623942480 Ack=4179456069 Win=65535 Len=0
1182 2019-10-22 17:49:03.874542 173.37.145.8   192.168.0.100  TCP        58                   443 → 39924 [RST, ACK] Seq=4179456069 Ack=623942480 Win=9952 Len=0

      Length: 5339
    › Handshake Protocol: Server Hello
    ⌄ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 5240
        Certificates Length: 5237
      ⌄ Certificates (5237 bytes)
          Certificate Length: 2025
        ⌄ Certificate: 308207e5308205cda00302010202143000683b0f7504f7b2... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose,
          › signedCertificate
          › algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted: 6921d084f7a6f6167058f14e2aad8b98b4e6c971ea6ea3b4...
          Certificate Length: 1736
        ⌄ Certificate: 308206c4308204aca00302010202147517167783d0437eb5... (id-at-commonName=HydrantID SSL ICA G2,id-at-organizationName=HydrantID (Avalanche Cloud Corporation),id-
          ⌄ 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
```

這些捕獲證明傳輸防火牆修改了伺服器證書(MITM)

行動2.檢查裝置日誌。

您可以收集本檔案中所述FMC TS套件組合：

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

在這種情況下，/dir-archives/var-log/process_stdout.log檔案會顯示以下訊息：

<#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/
```

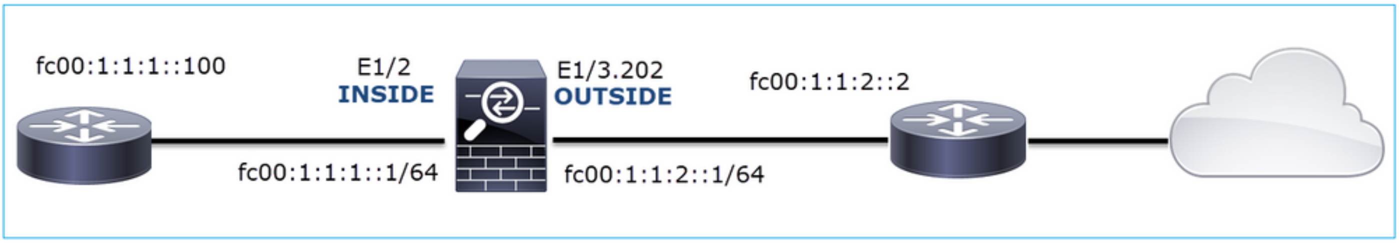
推薦的解決方案

禁用特定流的MITM，以便FMC可以成功註冊到智慧許可雲。

案例11.IPv6連線問題

問題描述：內部主機（位於防火牆的INSIDE介面之後）無法與外部主機（位於防火牆的OUTSIDE介面之後的主機）通訊。

下圖顯示拓撲：



受影響的流：

源IP:fc00:1:1:1::100

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

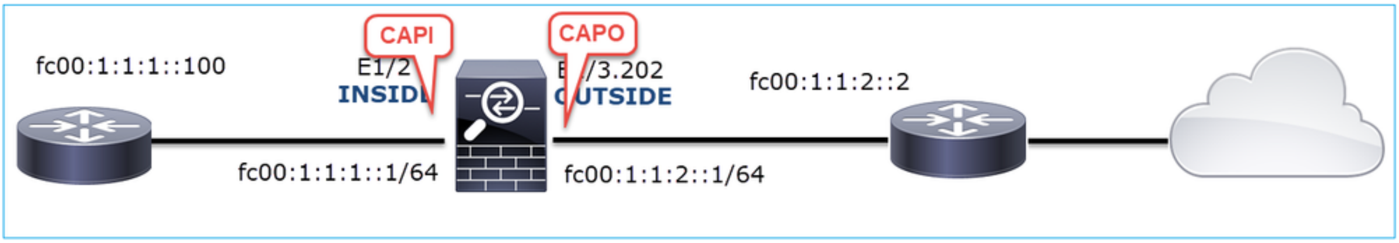
協定：任意

捕獲分析

在FTD LINA引擎上啟用擷取。

<#root>

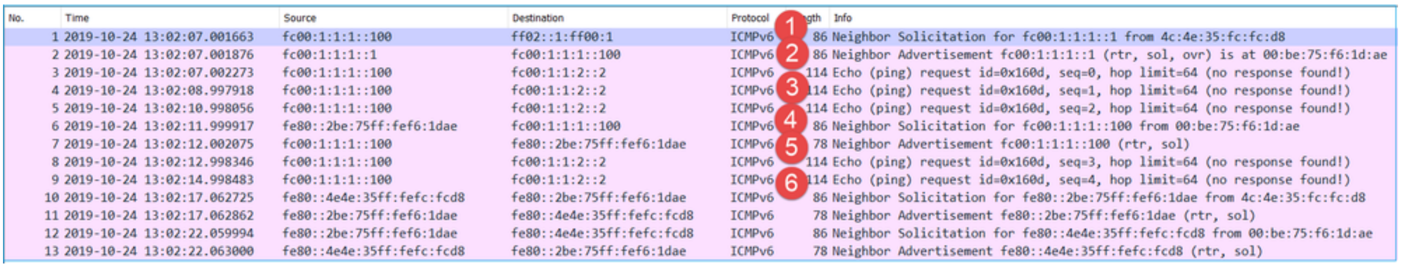
```
firepower#

capture CAPI int INSIDE match ip any6 any6

firepower#

capture CAPO int OUTSIDE match ip any6 any6
```



捕獲 — 非功能場景

這些捕獲與ICMP連線測試並行進行，ICMP連線測試從IP fc00:1:1:1::100（內部路由器）到IP fc00:1:1:2::2（上游路由器）。

防火牆INSIDE介面上的捕獲包含：

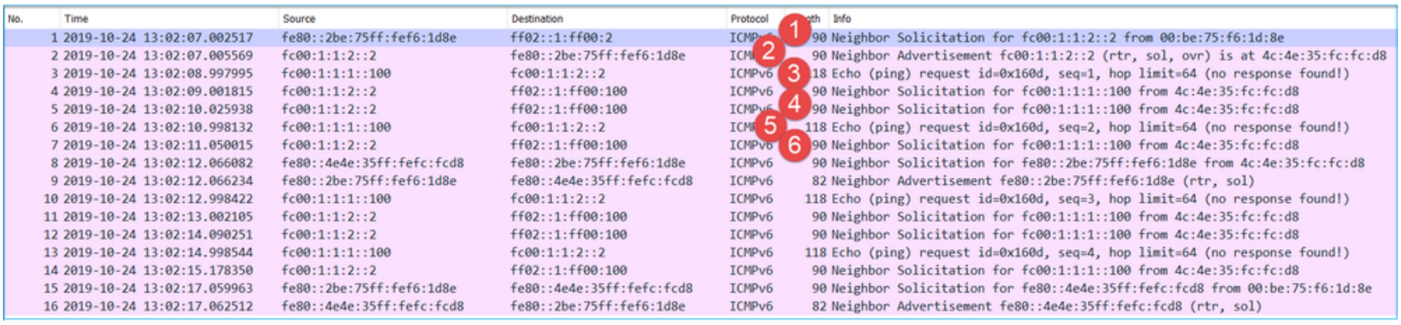


No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 13:02:07.001663	fc00:1:1:1::100	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
2	2019-10-24 13:02:07.001876	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6	86	Neighbor Advertisement fc00:1:1:1::1 (rtr, sol, ovr) 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)

重點：

1. 路由器傳送IPv6 Neighbor Solicitation消息並請求上游裝置的MAC地址(IP fc00:1:1:1::1)。
2. 防火牆使用IPv6鄰居通告進行響應。
3. 路由器傳送一個ICMP回應請求。
4. 防火牆傳送IPv6鄰居請求消息並請求下游裝置的MAC地址(fc00:1:1:1::100)。
5. 路由器使用IPv6鄰居通告進行應答。
6. 路由器會傳送額外的IPv6 ICMP回應請求。

防火牆OUTSIDE介面上的捕獲包含：



No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 13:02:07.002517	fe80::2be:75ff:fef6:1d8e	ff02::1:ff00:2	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
2	2019-10-24 13:02:07.005569	fc00:1:1:2::2	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
3	2019-10-24 13:02:08.997995	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	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)

重點：

1. 防火牆傳送IPv6鄰居請求消息，該消息要求輸入上游裝置的MAC地址(IP fc00:1:1:2::2)。
2. 路由器使用IPv6鄰居通告進行應答。
3. 防火牆會傳送IPv6 ICMP回應請求。
4. 上游裝置（路由器fc00:1:1:2::2）傳送IPv6鄰居請求消息，該消息要求獲取IPv6地址fc00:1:1:1::100的MAC地址。
5. 防火牆會傳送額外的IPv6 ICMP回應請求。
6. 上游路由器傳送一個額外的IPv6鄰居請求消息，該消息要求IPv6地址fc00:1:1:1::100的MAC地址。

第4點很有意思。通常，上游路由器請求防火牆外部介面(fc00:1:1:2::2)的MAC，但實際上它請求的是fc00:1:1:1::100。這表示組態錯誤。

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.檢查IPv6鄰居表。

防火牆IPv6鄰居表已正確填充。

<#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
```

行動2.檢查IPv6配置。

這是防火牆配置。

<#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
```

上游裝置配置顯示配置錯誤：

<#root>

```
Router#
```

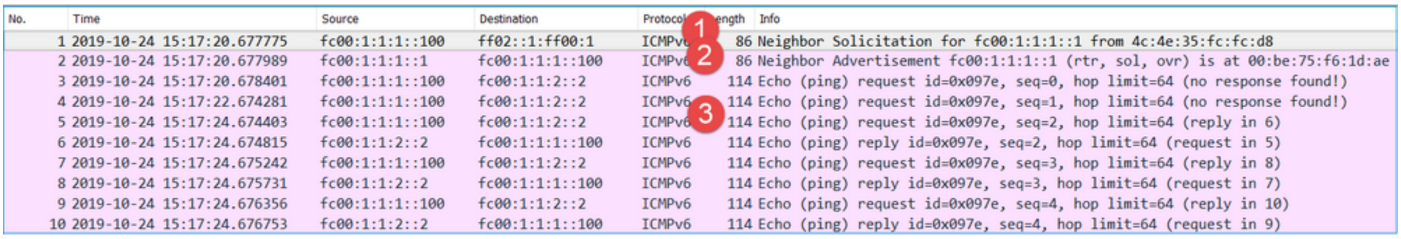
```
show run interface g0/0.202
```

```
!
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
```

捕獲 — 功能方案

子網掩碼更改（從/48更改為/64）解決了此問題。這是功能方案中的CAPI捕獲。

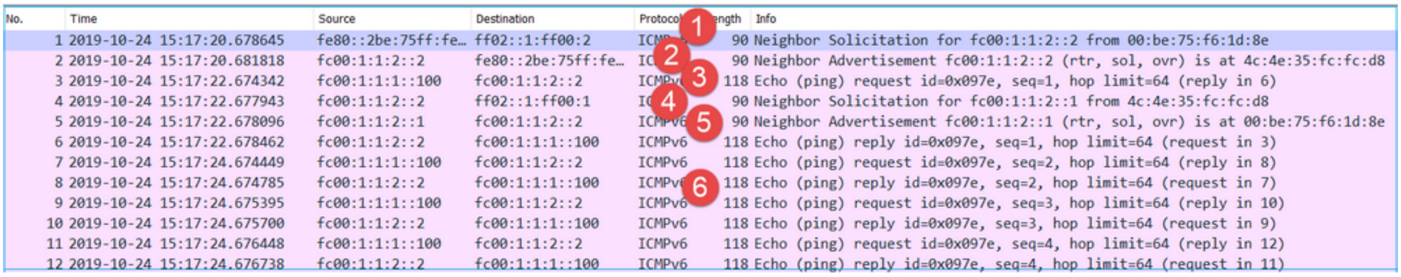


```
No.  Time                        Source            Destination       Protocol Length Info
   1 2019-10-24 15:17:20.677775  fc00:1:1:1::100   ff02::1:ff00:1    ICMPv6    86 Neighbor Solicitation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
   2 2019-10-24 15:17:20.677989  fc00:1:1:1::1     fc00:1:1:1::100   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)
```

要點：

1. 路由器傳送IPv6鄰居請求消息，該消息要求輸入上游裝置的MAC地址(IP fc00:1:1:1::1)。
2. 防火牆使用IPv6鄰居通告進行響應。
3. 路由器傳送ICMP回應請求並獲得回應回覆。

CAPO內容：



No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 15:17:20.678645	fe80::2be:75ff:fe…	ff02::1:ff00:2	ICMP	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…	IC	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	ICMPv	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	IC	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)

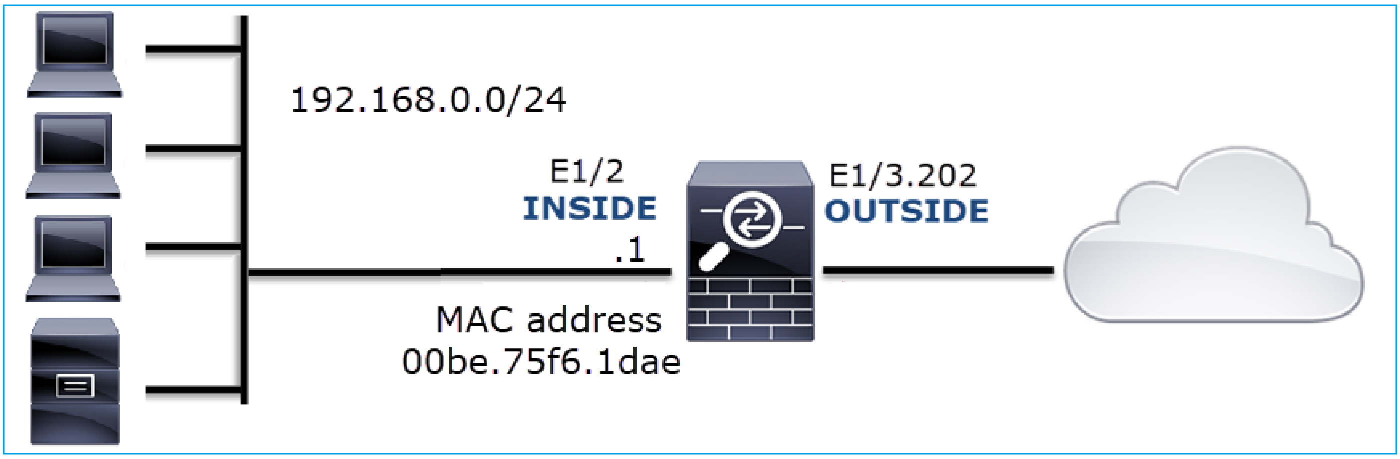
重點：

1. 防火牆傳送IPv6鄰居請求消息，該消息要求輸入上游裝置的MAC地址(IP fc00:1:1:2::2)。
2. 防火牆使用IPv6鄰居通告進行響應。
3. 防火牆傳送ICMP回應請求。
4. 路由器傳送一條IPv6鄰居請求消息，詢問下游裝置的MAC地址(IP fc00:1:1:1::1)。
5. 防火牆使用IPv6鄰居通告進行響應。
6. 防火牆會傳送ICMP回應要求並獲得回應回覆。

案例12.間歇性連線問題（ARP中毒）

問題描述：內部主機(192.168.0.x/24)與同一子網中的主機存在間歇性連線問題

下圖顯示拓撲：



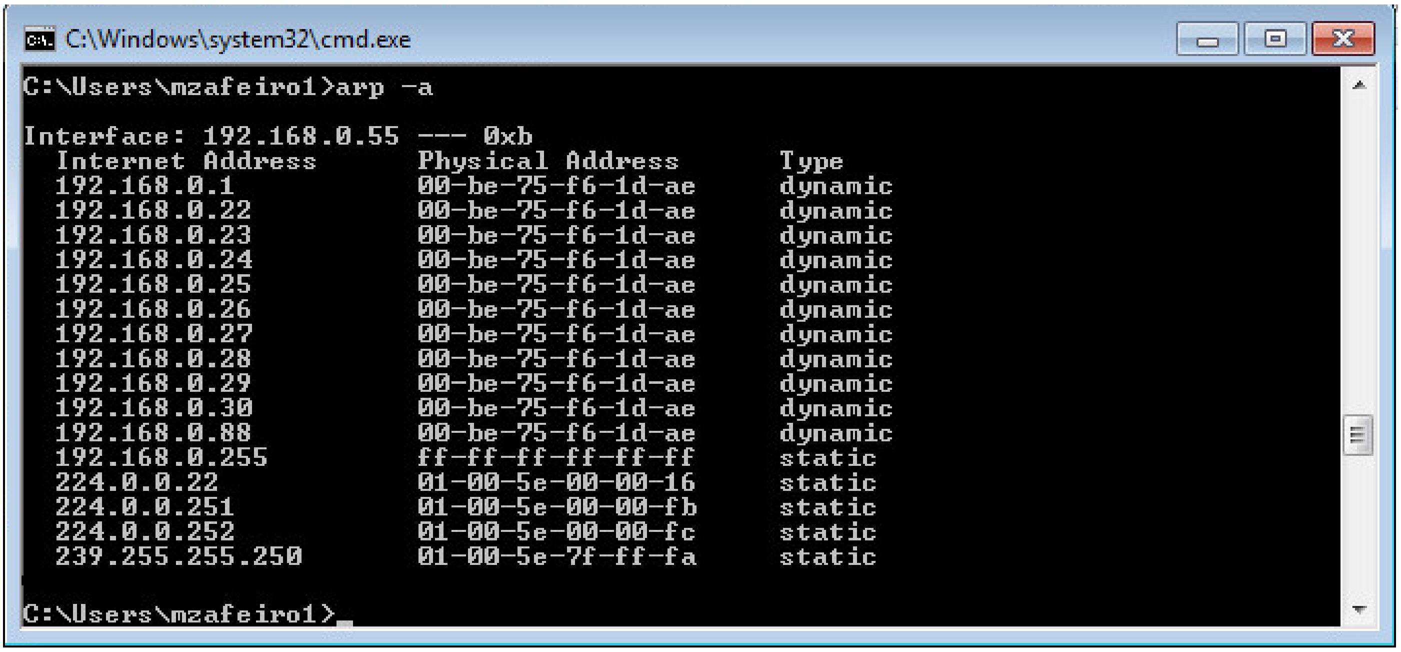
受影響的流：

源IP:192.168.0.x/24

Dst IP:192.168.0.x/24

協定：任意

內部主機的ARP快取似乎已中毒：



```
C:\Users\mzafeiro1>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\mzafeiro1>_
```

捕獲分析

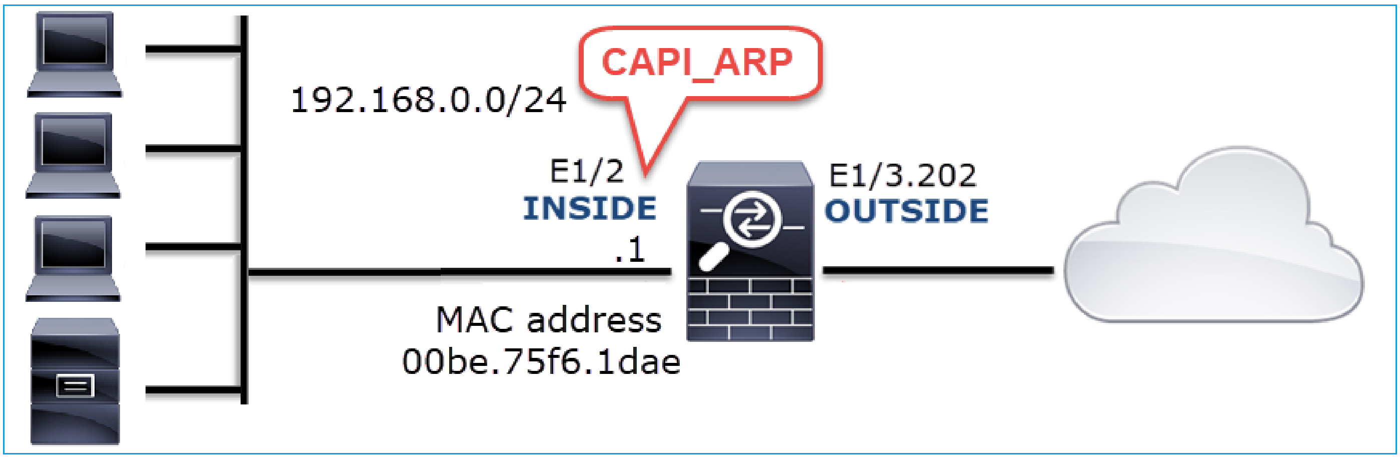
在FTD LINA引擎上啟用擷取

此擷取僅擷取INSIDE介面上的ARP封包：

<#root>

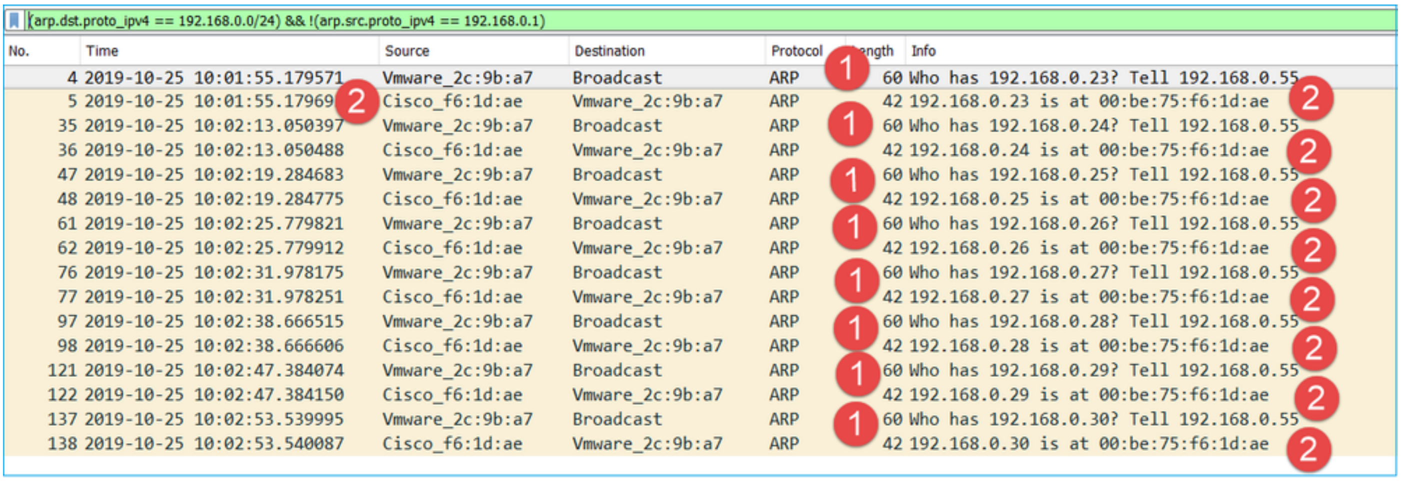
```
firepower#
```

```
capture CAPI_ARP interface INSIDE ethernet-type arp
```



捕獲 — 非功能場景：

防火牆INSIDE介面上的捕獲包含。



(arp.dst.proto_ipv4 == 192.168.0.0/24) && !(arp.src.proto_ipv4 == 192.168.0.1)

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

重點：

1. 防火牆接收192.168.0.x/24網路內IP的各種ARP請求
2. 防火牆會使用自己的MAC位址回應所有封包（代理ARP）

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.檢查NAT配置。

針對NAT配置，存在no-proxy-arp關鍵字可阻止早期行為的情況：

<#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
```

行動2.在防火牆介面上停用proxy-arp功能。

如果「no-proxy-arp」關鍵字不能解決問題，請嘗試在介面本身上停用代理ARP。如果是FTD，則在撰寫本文時，您必須使用FlexConfig並部署命令（指定適當的介面名稱）。

```
sysopt noproxyarp INSIDE
```

案例13.標識導致CPU佔用的SNMP對象識別符號(OID)

此案例展示，如何根據對SNMP第3版(SNMPv3)封包擷取的分析，將某些用於記憶體輪詢的SNMP OID識別為CPU存取（效能問題）的根本原因。

問題描述：資料介面上的超限持續增加。進一步的研究表明，也有CPU存取（由SNMP進程引起）是介面超載的根本原因。

故障排除過程的下一步是確定由SNMP進程引起的CPU佔用問題的根本原因，尤其是縮小問題的範圍，以確定SNMP對象識別符號(OID)，在輪詢時，OID可能會導致CPU佔用問題。

目前，FTD LINA引擎不會為即時輪詢的SNMP OID提供「show」命令。

用於輪詢的SNMP OID清單可以從SNMP監控工具中檢索，但是在這種情況下，存在以下預防因素：

- FTD管理員無法存取SNMP監控工具
- 在FTD上設定了具有隱私驗證和資料加密的SNMP第3版

捕獲分析

由於FTD管理員具有SNMP第3版身份驗證和資料加密的憑證，因此建議採取以下措施：

1. 捕獲SNMP資料包捕獲
2. 儲存捕獲，並使用Wireshark SNMP協定首選項指定SNMP第3版憑證以解密SNMP第3版資料包。解密的捕獲用於分析和檢索SNMP OID

在用於snmp-server host配置的介面上配置SNMP資料包捕獲：

<#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 capsnmp interface management buffer 10000000 match udp host 192.168.10.10  host 192.168.5.254 ec
```

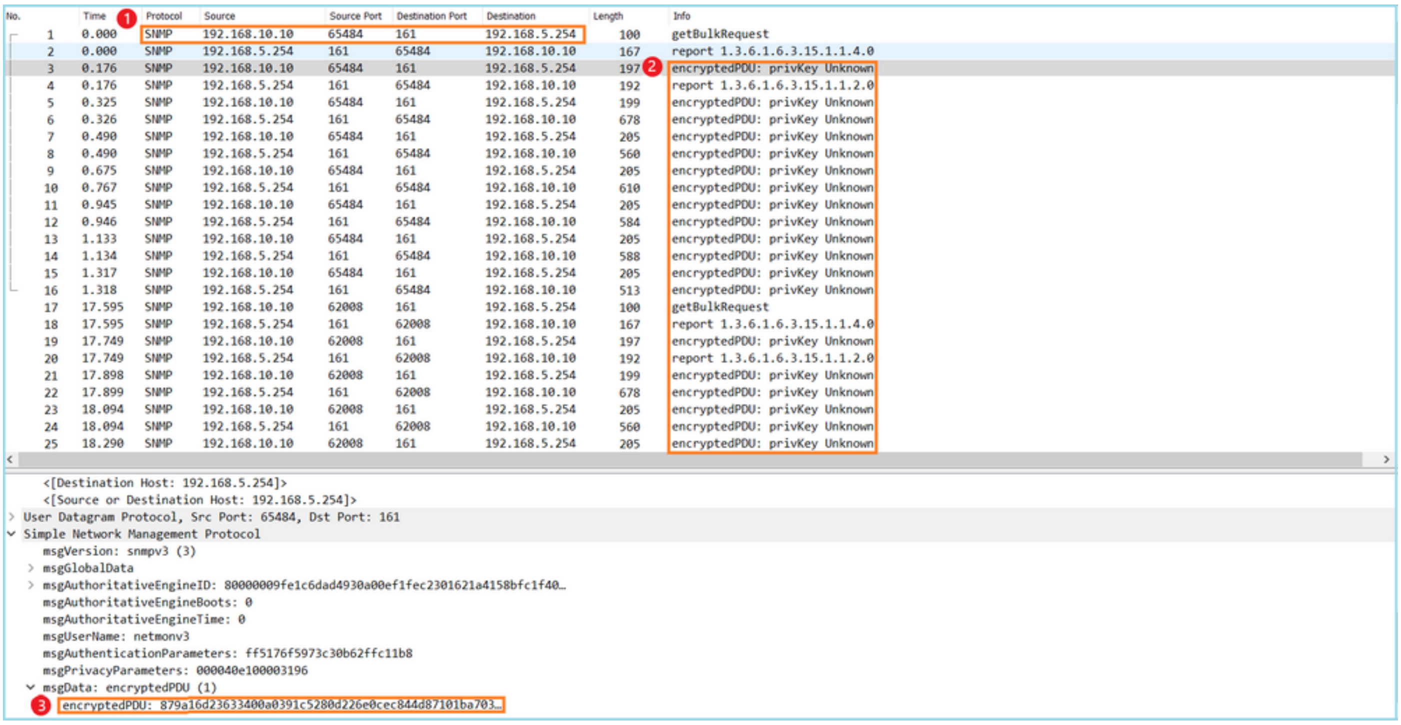
```
firepower#
```

```
show capture capsnmp
```

```
capture capsnmp 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
v Simple Network Management Protocol
    msgVersion: snmpv3 (3)
  > msgGlobalData
  > msgAuthoritativeEngineID: 80000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40…
    msgAuthoritativeEngineBoots: 0
    msgAuthoritativeEngineTime: 0
    msgUserName: netmonv3
    msgAuthenticationParameters: ff5176f5973c30b62ffc11b8
    msgPrivacyParameters: 000040e100003196
  v msgData: encryptedPDU (1)
      encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703…
```

重點：

1. SNMP源地址和目的地地址/埠。
2. 無法解碼SNMP協定PDU，因為privKey對Wireshark未知。
3. encryptedPDU基元的值。

建議的操作

本節所列的行動旨在進一步縮小問題範圍。

操作1.解密SNMP捕獲。

儲存捕獲並編輯Wireshark SNMP協定首選項以指定SNMP版本3憑證以解密資料包。

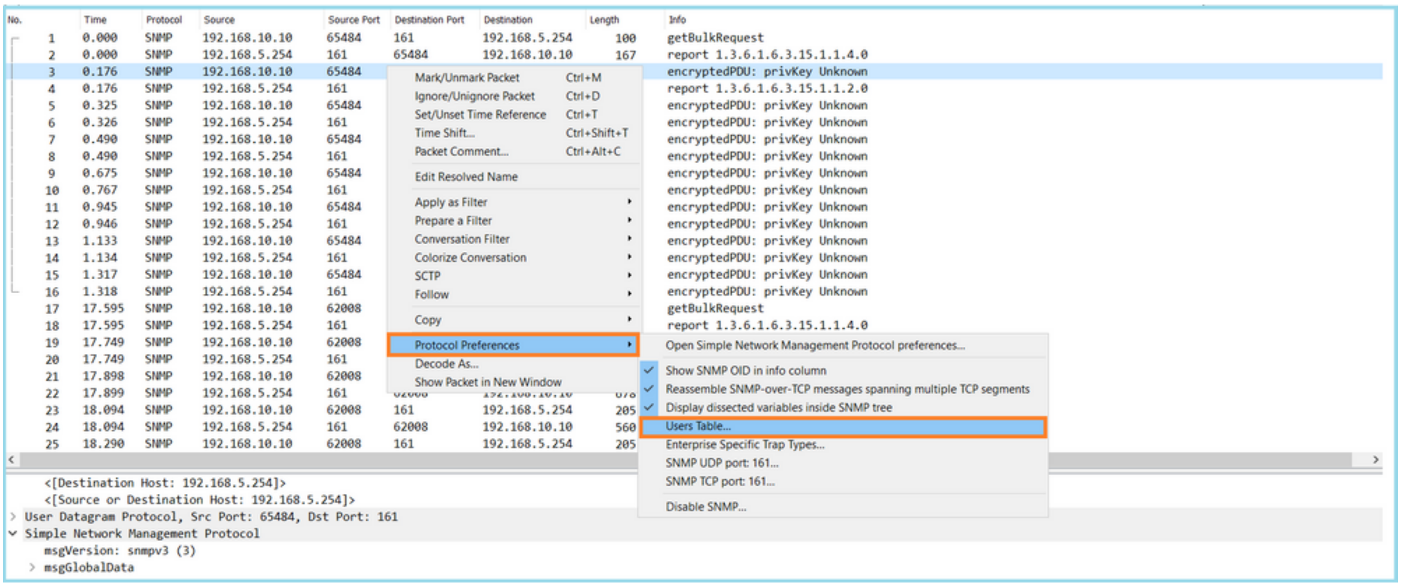
<#root>

```
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
```

在Wireshark上開啟捕獲檔案，選擇一個SNMP資料包並導航到Protocol Preferences > Users Table，如下圖所示：

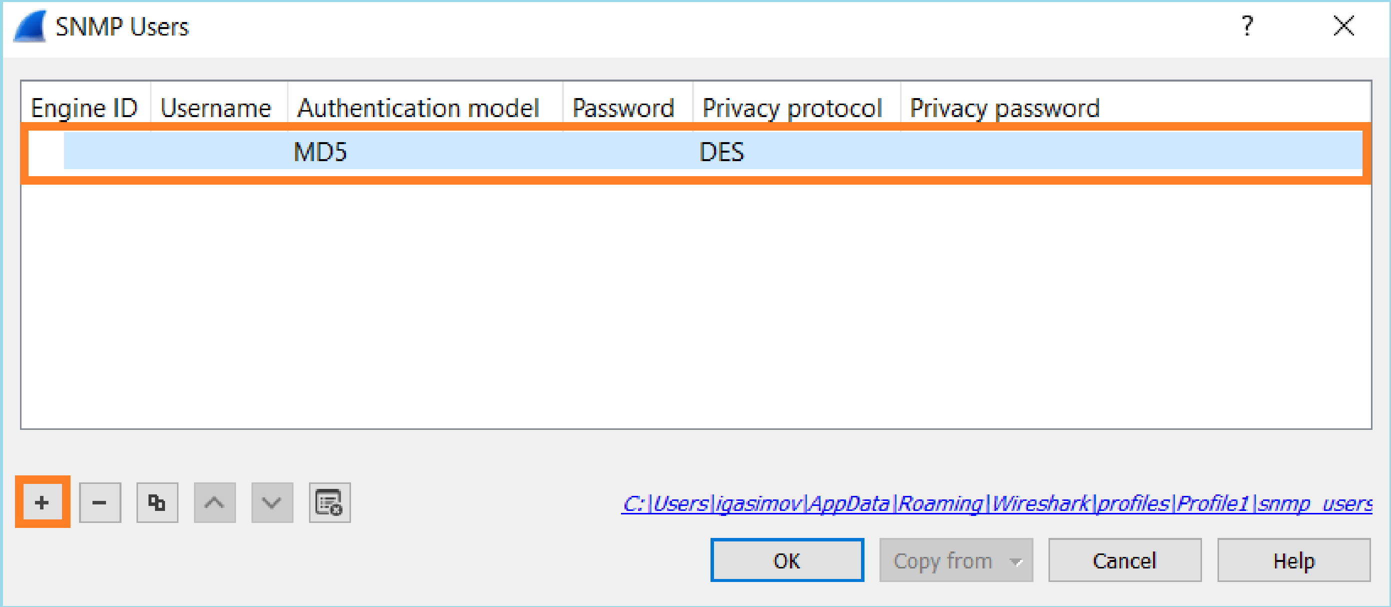


No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
3	0.176	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
4	0.176	SNMP	192.168.5.254	161				report 1.3.6.1.6.3.15.1.1.2.0
5	0.325	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
6	0.326	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
7	0.490	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
8	0.490	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
9	0.675	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
10	0.767	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
11	0.945	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
12	0.946	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
13	1.133	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
14	1.134	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
15	1.317	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
16	1.318	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
17	17.595	SNMP	192.168.10.10	62008				getBulkRequest
18	17.595	SNMP	192.168.5.254	161				report 1.3.6.1.6.3.15.1.1.4.0
19	17.749	SNMP	192.168.10.10	62008				
20	17.749	SNMP	192.168.5.254	161				
21	17.898	SNMP	192.168.10.10	62008				
22	17.899	SNMP	192.168.5.254	161				
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	

```
    <[Destination Host: 192.168.5.254]>
    <[Source or Destination Host: 192.168.5.254]>
> User Datagram Protocol, Src Port: 65484, Dst Port: 161
v Simple Network Management Protocol
    msgVersion: snmpv3 (3)
  > msgGlobalData
```

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
Open Simple Network Management Protocol preferences...
✓ Show SNMP OID in info column
✓ Reassemble SNMP-over-TCP messages spanning multiple TCP segments
✓ Display dissected variables inside SNMP tree
Users Table...
Enterprise Specific Trap Types...
SNMP UDP port: 161...
SNMP TCP port: 161...
Disable SNMP...

在SNMP Users表中，指定了SNMP版本3使用者名稱、身份驗證模型、身份驗證密碼、隱私協定和隱私密碼（下面未顯示實際憑據）：

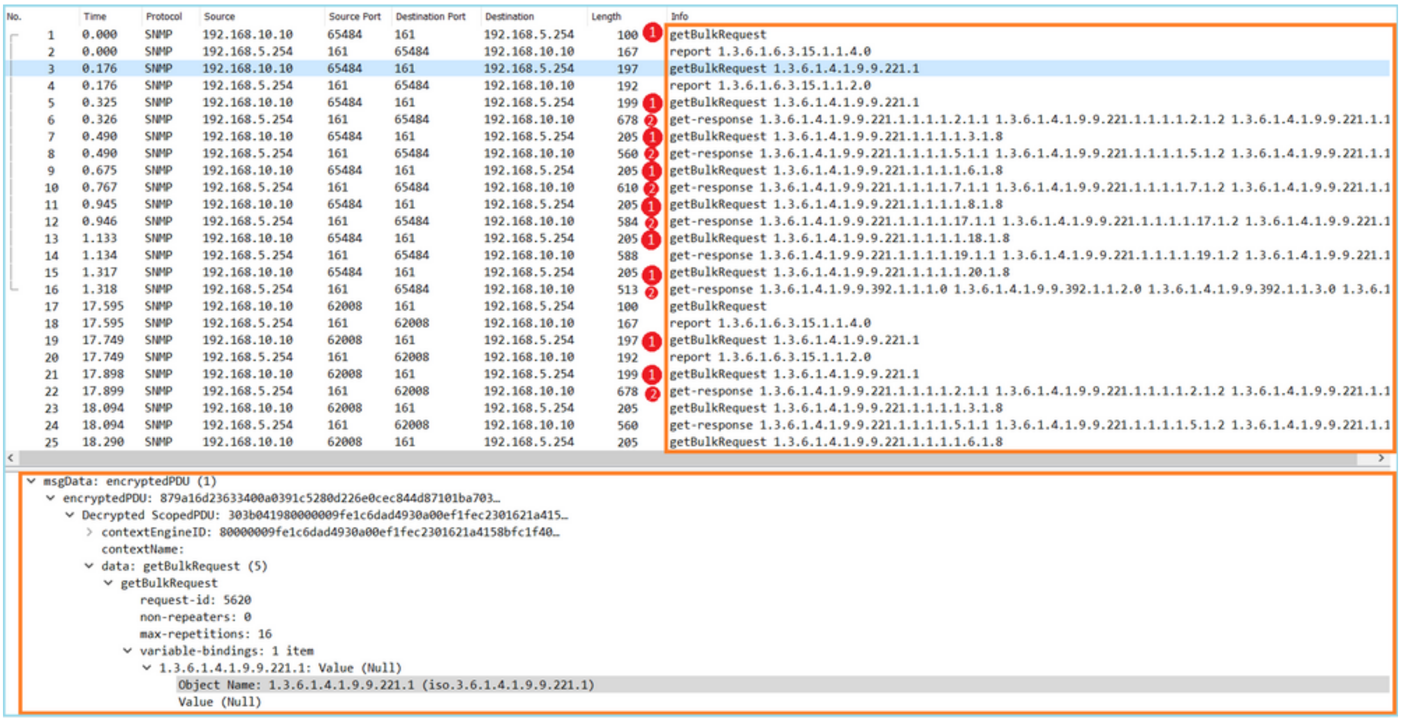


SNMP Users

Engine ID	Username	Authentication model	Password	Privacy protocol	Privacy password
		MD5		DES	

C:\Users\igasimov\AppData\Roaming\Wireshark\profiles\Profile1\snmp_users

應用SNMP使用者設定後，Wireshark顯示已解密的SNMP PDU:



```
No.  Time    Protocol Source         Source Port Destination Port Destination    Length Info
1    0.000   SNMP     192.168.10.10  65484       161              192.168.5.254  100    getBulkRequest
2    0.000   SNMP     192.168.5.254  161         65484            192.168.10.10  167    report 1.3.6.1.6.3.15.1.1.4.0
3    0.176   SNMP     192.168.10.10  65484       161              192.168.5.254  197    getBulkRequest 1.3.6.1.4.1.9.9.221.1
4    0.176   SNMP     192.168.5.254  161         65484            192.168.10.10  192    report 1.3.6.1.6.3.15.1.1.2.0
5    0.325   SNMP     192.168.10.10  65484       161              192.168.5.254  199    getBulkRequest 1.3.6.1.4.1.9.9.221.1
6    0.326   SNMP     192.168.5.254  161         65484            192.168.10.10  678    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1
7    0.490   SNMP     192.168.10.10  65484       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8
8    0.490   SNMP     192.168.5.254  161         65484            192.168.10.10  560    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1
9    0.675   SNMP     192.168.10.10  65484       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8
10   0.767   SNMP     192.168.5.254  161         65484            192.168.10.10  610    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.7.1.2 1.3.6.1.4.1.9.9.221.1.1
11   0.945   SNMP     192.168.10.10  65484       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.8.1.8
12   0.946   SNMP     192.168.5.254  161         65484            192.168.10.10  584    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.17.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.17.1.2 1.3.6.1.4.1.9.9.221.1
13   1.133   SNMP     192.168.10.10  65484       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.18.1.8
14   1.134   SNMP     192.168.5.254  161         65484            192.168.10.10  588    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.19.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.19.1.2 1.3.6.1.4.1.9.9.221.1
15   1.317   SNMP     192.168.10.10  65484       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.20.1.8
16   1.318   SNMP     192.168.5.254  161         65484            192.168.10.10  513    get-response 1.3.6.1.4.1.9.9.392.1.1.1.0 1.3.6.1.4.1.9.9.392.1.1.2.0 1.3.6.1.4.1.9.9.392.1.1.3.0 1.3.6.1
17   17.595  SNMP     192.168.10.10  62008       161              192.168.5.254  100    getBulkRequest
18   17.595  SNMP     192.168.5.254  161         62008            192.168.10.10  167    report 1.3.6.1.6.3.15.1.1.4.0
19   17.749  SNMP     192.168.10.10  62008       161              192.168.5.254  197    getBulkRequest 1.3.6.1.4.1.9.9.221.1
20   17.749  SNMP     192.168.5.254  161         62008            192.168.10.10  192    report 1.3.6.1.6.3.15.1.1.2.0
21   17.898  SNMP     192.168.10.10  62008       161              192.168.5.254  199    getBulkRequest 1.3.6.1.4.1.9.9.221.1
22   17.899  SNMP     192.168.5.254  161         62008            192.168.10.10  678    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1
23   18.094  SNMP     192.168.10.10  62008       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8
24   18.094  SNMP     192.168.5.254  161         62008            192.168.10.10  560    get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1
25   18.290  SNMP     192.168.10.10  62008       161              192.168.5.254  205    getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8
```

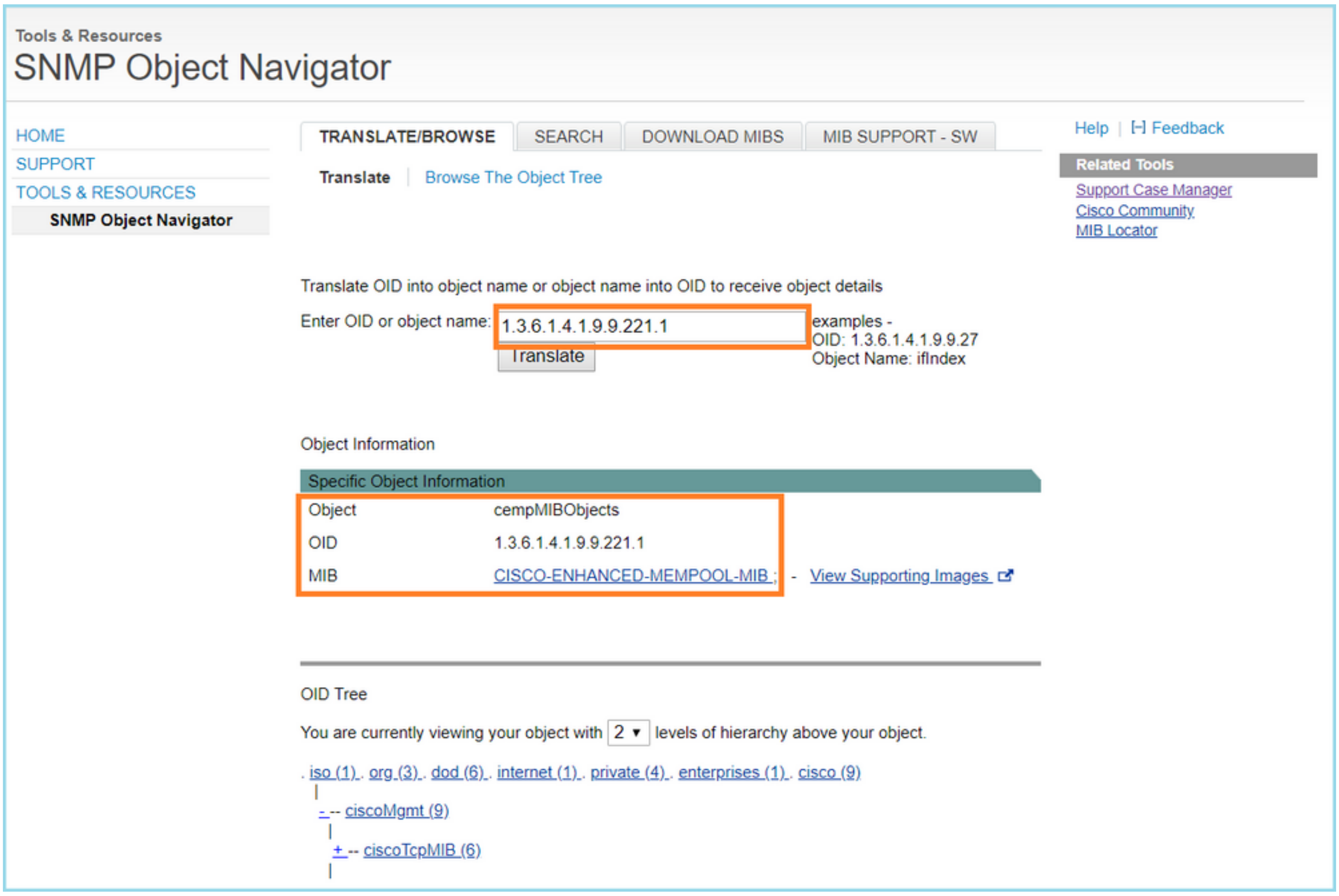
```
v msgData: encryptedPDU (1)
  v encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703…
    v Decrypted ScopedPDU: 303b041980000009fe1c6dad4930a00ef1fec2301621a415…
      > contextEngineID: 80000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40…
        contextName:
      v data: getBulkRequest (5)
        v getBulkRequest
            request-id: 5620
            non-repeaters: 0
            max-repetitions: 16
          v variable-bindings: 1 item
            v 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)
```

重點：

1. SNMP監控工具使用SNMP getBulkRequest查詢和遍歷父OID 1.3.6.1.4.1.9.9.221.1和相關OID。
2. FTD透過包含與1.3.6.1.4.1.9.9.221.1相關的OID的get-response回應每個getBulkRequest。

行動2.識別SNMP OID。

SNMP物件導覽器顯示OID 1.3.6.1.4.1.9.9.221.1屬於名為CISCO-ENHANCED-MEMPOOL-MIB的管理資訊庫(MIB)，如下圖所示：



Translate OID into object name or object name into OID to receive object details

Enter OID or object name: 1.3.6.1.4.1.9.9.221.1

examples -
OID: 1.3.6.1.4.1.9.9.27
Object Name: ifIndex

Object Information

Specific Object Information	
Object	cempMIBObjects
OID	1.3.6.1.4.1.9.9.221.1
MIB	CISCO-ENHANCED-MEMPOOL-MIB ; - View Supporting Images

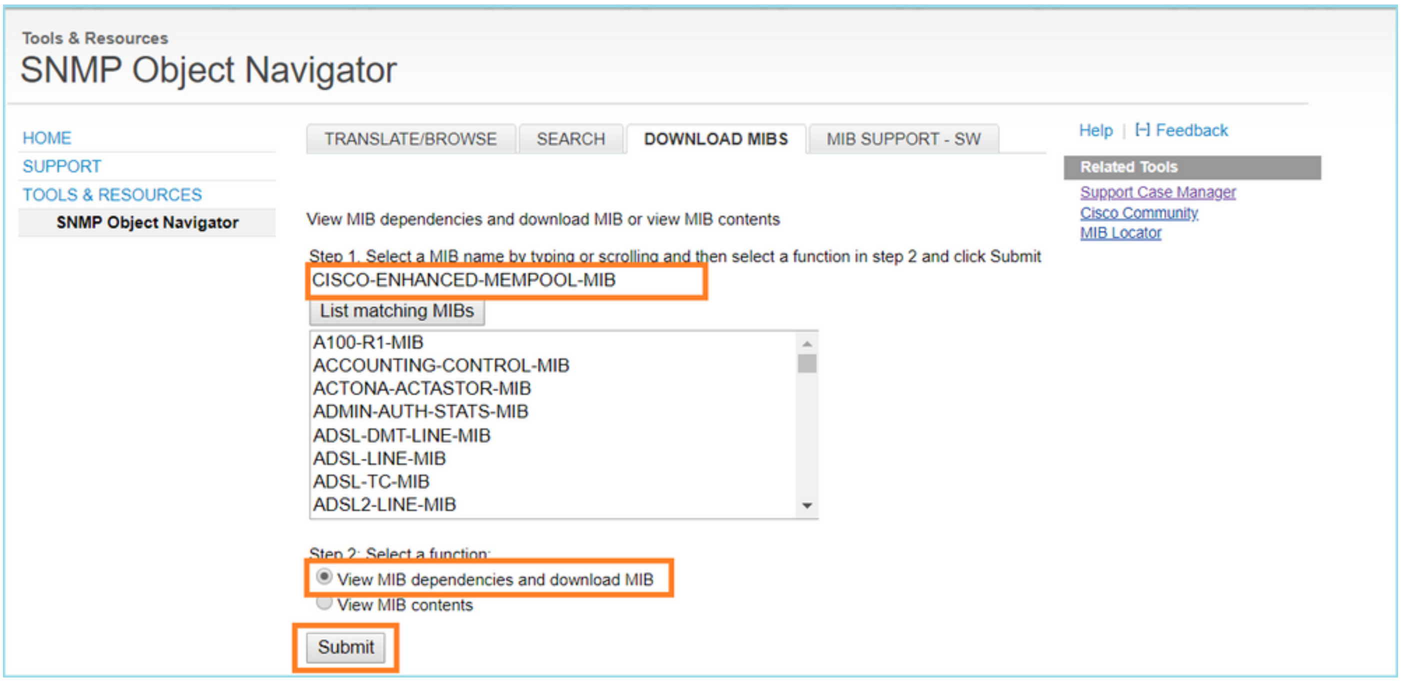
OID Tree

You are currently viewing your object with 2 ▾ levels of hierarchy above your object.

```
. iso (1) . org (3) . dod (6) . internet (1) . private (4) . enterprises (1) . cisco (9)
  |
  - -- ciscoMgmt (9)
    |
    + -- ciscoTcpMIB (6)
    |
```

要在Wireshark中以可讀格式顯示OID，請執行以下操作：

1. 下載MIB CISCO-ENHANCED-MEMPOOL-MIB及其依賴項，如下圖所示：



View MIB dependencies and download MIB or view MIB contents

Step 1. Select a MIB name by typing or scrolling and then select a function in step 2 and click Submit

CISCO-ENHANCED-MEMPOOL-MIB

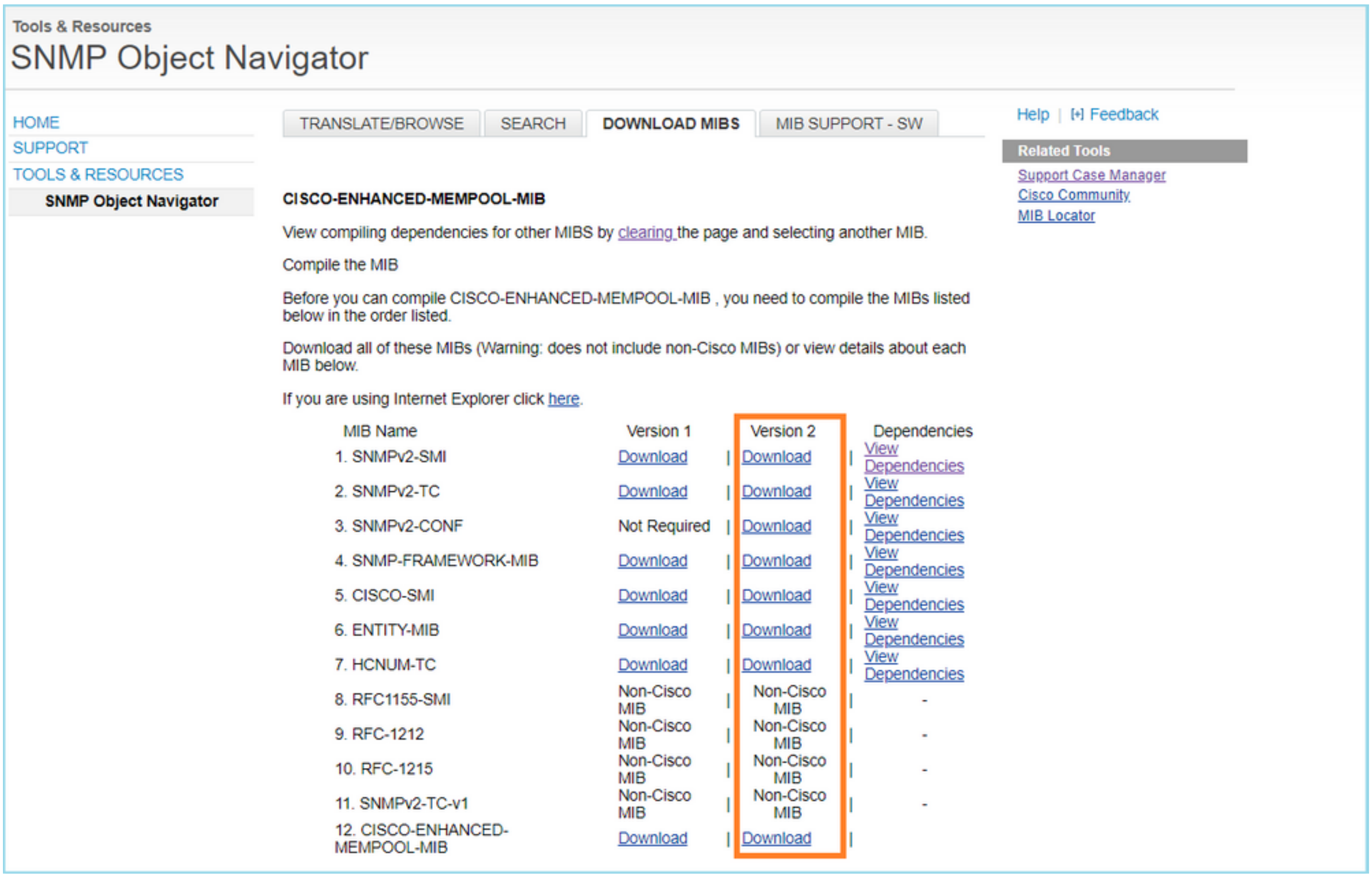
List matching MIBs

A100-R1-MIB
ACCOUNTING-CONTROL-MIB
ACTONA-ACTASTOR-MIB
ADMIN-AUTH-STATS-MIB
ADSL-DMT-LINE-MIB
ADSL-LINE-MIB
ADSL-TC-MIB
ADSL2-LINE-MIB

Step 2. Select a function:

● View MIB dependencies and download MIB
○ View MIB contents

Submit



Tools & Resources

SNMP Object Navigator

CISCO-ENHANCED-MEMPOOL-MIB

View compiling dependencies for other MIBS by clearing the page and selecting another MIB.

Compile the MIB

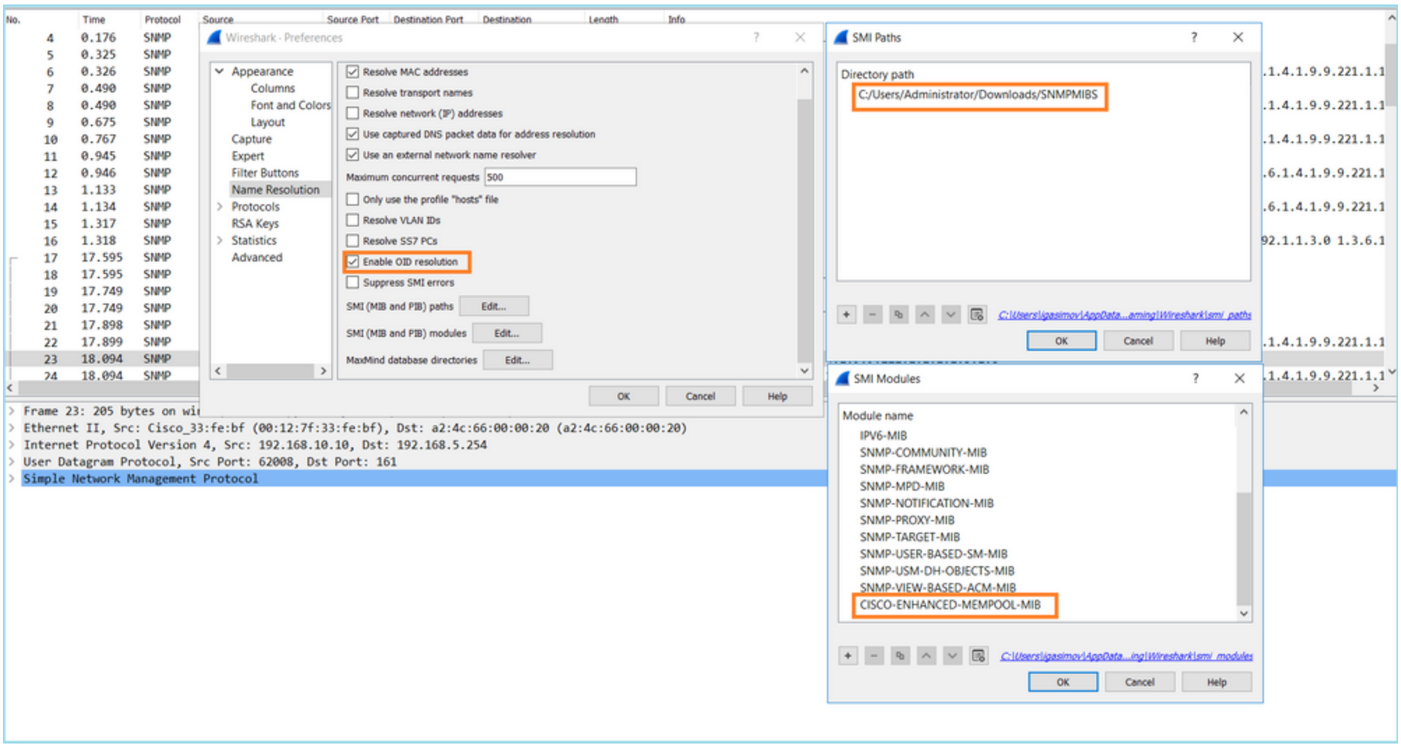
Before you can compile CISCO-ENHANCED-MEMPOOL-MIB , you need to compile the MIBs listed below in the order listed.

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

If you are using Internet Explorer click here.

MIB Name	Version 1	Version 2	Dependencies
1. SNMPv2-SMI	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.在Wireshark的編輯>首選項>名稱解析視窗中，選中啟用OID解析。在SMI（MIB和PIB路徑）窗口中，使用下載的MIB和SMI（MIB和PIB模組）指定資料夾。CISCO-ENHANCED-MEMPOOL-MIB會自動新增到模組清單中：



No.	Time	Protocol
4	0.176	SNMP
5	0.325	SNMP
6	0.326	SNMP
7	0.490	SNMP
8	0.490	SNMP
9	0.675	SNMP
10	0.767	SNMP
11	0.945	SNMP
12	0.946	SNMP
13	1.133	SNMP
14	1.134	SNMP
15	1.317	SNMP
16	1.318	SNMP
17	17.595	SNMP
18	17.595	SNMP
19	17.749	SNMP
20	17.749	SNMP
21	17.898	SNMP
22	17.899	SNMP
23	18.094	SNMP
24	18.094	SNMP

```
.1.4.1.9.9.221.1.1
.1.4.1.9.9.221.1.1
.1.4.1.9.9.221.1.1
.6.1.4.1.9.9.221.1
.6.1.4.1.9.9.221.1
92.1.1.3.0 1.3.6.1
.1.4.1.9.9.221.1.1
.1.4.1.9.9.221.1.1
```

```
> Frame 23: 205 bytes on wi
> Ethernet II, Src: Cisco_33:fe:bf (00:12:7f:33:fe:bf), Dst: a2:4c:66:00:00:20 (a2:4c:66:00:00:20)
> Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.5.254
> User Datagram Protocol, Src Port: 62008, Dst Port: 161
> Simple Network Management Protocol
```

Wireshark · Preferences

- ☑ Resolve MAC addresses
- □ Resolve transport names
- □ Resolve network (IP) addresses
- ☑ Use captured DNS packet data for address resolution
- ☑ Use an external network name resolver
- Maximum concurrent requests 500
- □ Only use the profile "hosts" file
- □ Resolve VLAN IDs
- □ Resolve SS7 PCs
- ☑ Enable OID resolution
- □ Suppress SMI errors
- SMI (MIB and PIB) paths Edit...
- SMI (MIB and PIB) modules Edit...
- MaxMind database directories Edit...

SMI Paths

Directory path

C:/Users/Administrator/Downloads/SNMPMIBS

C:\Users\igasimov\AppData...aming\Wireshark\smi_paths

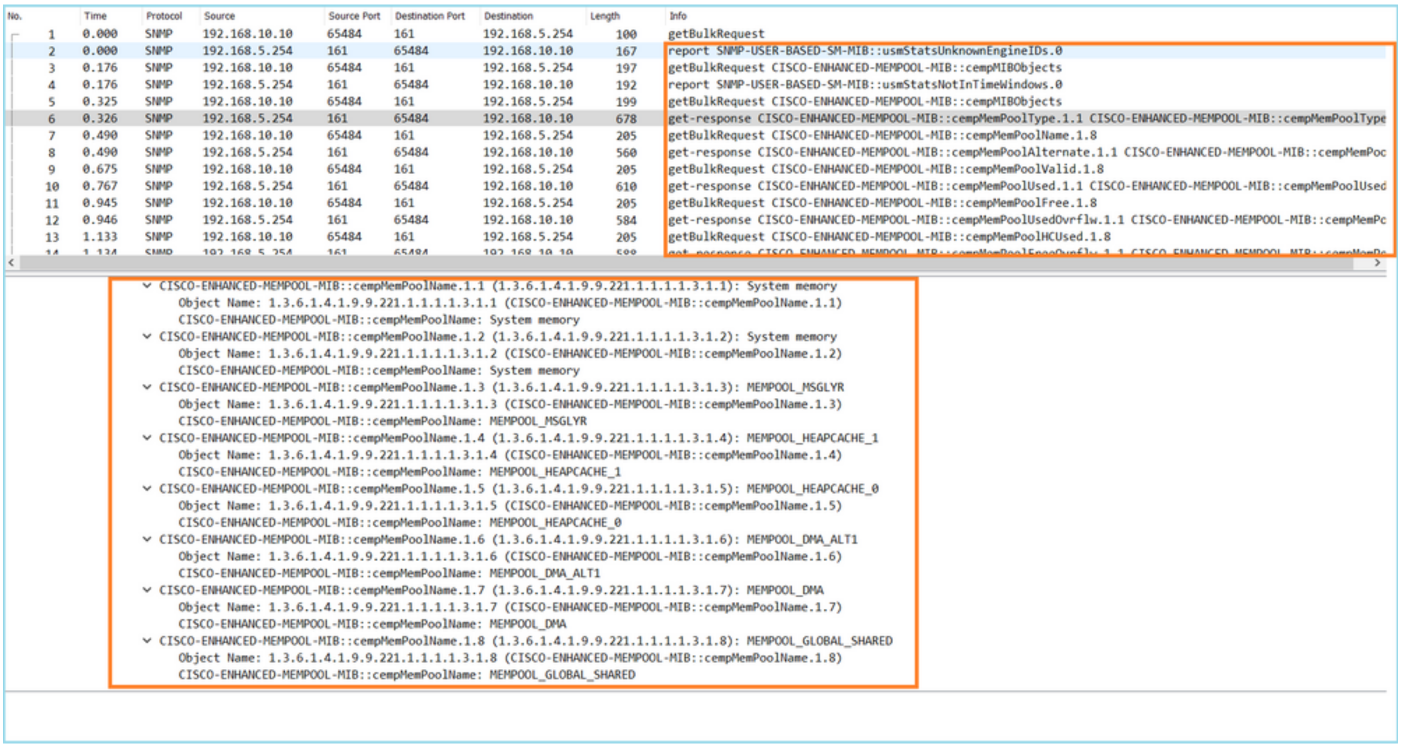
SMI Modules

Module name

IPV6-MIB
SNMP-COMMUNITY-MIB
SNMP-FRAMEWORK-MIB
SNMP-MPD-MIB
SNMP-NOTIFICATION-MIB
SNMP-PROXY-MIB
SNMP-TARGET-MIB
SNMP-USER-BASED-SM-MIB
SNMP-USM-DH-OBJECTS-MIB
SNMP-VIEW-BASED-ACM-MIB
CISCO-ENHANCED-MEMPOOL-MIB

C:\Users\igasimov\AppData...ing\Wireshark\smi_modules

3.重新啟動Wireshark後，OID解析將啟用：



No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report SNMP-USER-BASED-SM-MIB::usmStatsUnknownEngineIDs.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMIBObjects
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report SNMP-USER-BASED-SM-MIB::usmStatsNotInTimeWindows.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMIBObjects
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolType.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolType
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolAlternate.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoc
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolValid.1.8
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUsed.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUsed
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolFree.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUsedOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPc
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolHCUsed.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolFreeOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::cempMemPc

```
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.1 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.1): System memory
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.1 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.1)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: System memory
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.2 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.2): System memory
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.2 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.2)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: System memory
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.3 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.3): MEMPOOL_MSGLYR
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.3 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.3)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_MSGLYR
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.4 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.4): MEMPOOL_HEAPCACHE_1
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.4 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.4)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_HEAPCACHE_1
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.5 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.5): MEMPOOL_HEAPCACHE_0
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.5 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.5)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_HEAPCACHE_0
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.6 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.6): MEMPOOL_DMA_ALT1
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.6 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.6)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_DMA_ALT1
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.7 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.7): MEMPOOL_DMA
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.7 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.7)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_DMA
v CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.8 (1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8): MEMPOOL_GLOBAL_SHARED
    Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8 (CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName.1.8)
    CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolName: MEMPOOL_GLOBAL_SHARED
```

根據捕獲檔案的解密輸出，SNMP監控工具會定期（10秒間隔）輪詢有關FTD上記憶體池利用率的資料。如TechNote文章ASA SNMP Polling for Memory-Related Statistics中所述，使用SNMP輪詢全域性共用池(GSP)利用率會導致高CPU使用率。在本例中，從捕獲中可明顯看出，作為SNMP getBulkRequest基元的一部分，已定期輪詢全域性共用池利用率。

為了將SNMP進程導致的CPU佔用減至最低，建議遵循文章中提到的SNMP的CPU佔用緩解步驟，並避免輪詢與GSP相關的OID。如果不對與GSP相關的OID進行SNMP輪詢，則不會觀察到由SNMP進程導致的CPU佔用，並且超支率顯著降低。

相關資訊

- Cisco Firepower管理中心配置指南
- 釐清 Firepower Threat Defense 存取控制原則規則動作
- 使用Firepower威脅防禦捕獲和Packet Tracer
- 瞭解Wireshark

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。