

Análisis de las capturas de firewall de Firepower para solucionar problemas de red de manera eficaz

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[¿Cómo se recopilan y exportan capturas en la familia de productos NGFW?](#)

[Recopilar capturas de FXOS](#)

[Habilitar y recopilar capturas de línea de FTD](#)

[Activar y recopilar capturas de Snort de FTD](#)

[Troubleshoot](#)

[Caso 1. Sin TCP SYN en la interfaz de salida](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Resumen de posibles causas y acciones recomendadas](#)

[Caso 2. TCP SYN del cliente. TCP RST del servidor](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 3. Protocolo de enlace TCP de 3 vías + RST desde un terminal](#)

[Análisis de captura](#)

[3.1 - Protocolo de enlace TCP de 3 vías + RST retrasado del cliente](#)

[Acciones recomendadas](#)

[3.2 - Protocolo de enlace TCP de 3 vías + FIN/ACK retrasado del cliente + RST retrasado del servidor](#)

[Acciones recomendadas](#)

[3.3 - Protocolo de enlace TCP de 3 vías + RST retrasado del cliente](#)

[Acciones recomendadas](#)

[3.4 - Protocolo de enlace TCP de 3 vías + RST inmediato desde el servidor](#)

[Acciones recomendadas](#)

[Caso 4. TCP RST desde el cliente](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 5. Transferencia TCP lenta \(situación 1\)](#)

[Escenario 1. Transferencia lenta](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Situación hipotética 2. Transferencia rápida](#)

[Caso 6. Transferencia TCP lenta \(situación 2\)](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 7. Problema de conectividad TCP \(corrupción de paquetes\)](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 8. Problema de conectividad UDP \(paquetes faltantes\)](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 9. Problema de conectividad HTTPS \(situación 1\)](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 10. Problema de conectividad HTTPS \(situación 2\)](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 11. Problema de conectividad IPv6](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 12. Problema de conectividad intermitente \(envenenamiento ARP\)](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Caso 13. Identificar identificadores de objeto SNMP \(OID\) que provocan bloqueos de CPU](#)

[Análisis de captura](#)

[Acciones recomendadas](#)

[Información Relacionada](#)

Introducción

Este documento describe varias técnicas de análisis de captura de paquetes que tienen como objetivo resolver problemas de red de manera eficaz.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Arquitectura de la plataforma Firepower
- Registros de NGFW
- Rastreador de paquetes de NGFW

Además, antes de empezar a analizar las capturas de paquetes, es muy recomendable cumplir estos requisitos:

- Conozca la operación del protocolo - No comience a verificar una captura de paquetes si no entiende cómo funciona el protocolo capturado.
- Conozca la topología: debe conocer los dispositivos de tránsito de extremo a extremo. Si esto no es posible, debe conocer al menos los dispositivos de flujo ascendente y descendente.
- Conozca el dispositivo: debe saber cómo gestiona los paquetes su dispositivo, cuáles son las interfaces implicadas (entrada/salida), cuál es la arquitectura del dispositivo y cuáles son

los distintos puntos de captura.

- Conozca la configuración - Debe saber cómo se supone que el flujo de paquetes debe ser manejado por el dispositivo en términos de:
 - Interfaz de enrutamiento/salida
 - Políticas aplicadas
 - traducción de Dirección de Red (NAT)
- Conozca las herramientas disponibles - Junto con las capturas, se recomienda estar listo para aplicar otras herramientas y técnicas (como registro y trazadores) y, si es necesario, correlacionarlos con los paquetes capturados

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- La mayoría de las situaciones se basan en FP4140 con software FTD 6.5.x.
- FMC con software 6.5.x.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

La captura de paquetes es una de las herramientas de solución de problemas más pasadas por alto disponibles en la actualidad. A diario, Cisco TAC resuelve muchos problemas con el análisis de los datos capturados.

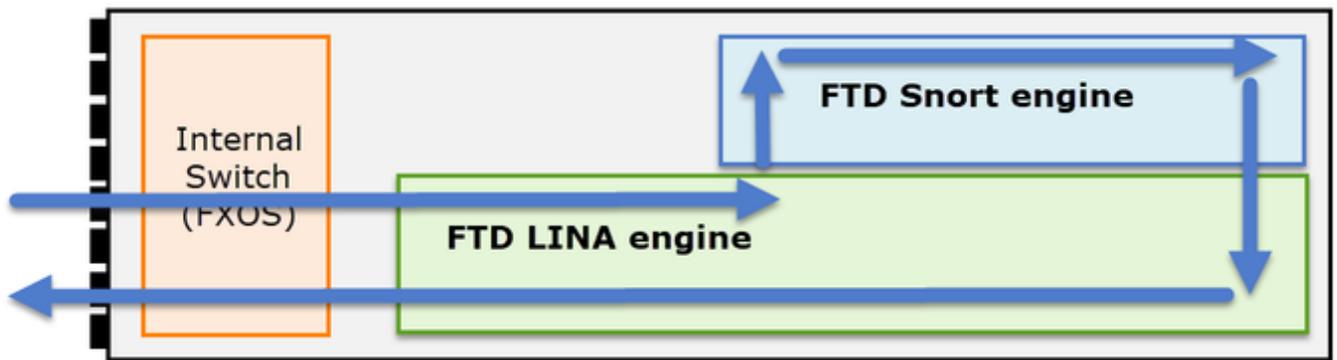
El objetivo de este documento es ayudar a los ingenieros de red y seguridad a identificar y resolver problemas comunes de red basados principalmente en el análisis de captura de paquetes.

Todos los escenarios presentados en este documento se basan en casos reales de usuarios vistos en el centro de asistencia técnica Cisco Technical Assistance Center (TAC).

El documento trata sobre las capturas de paquetes desde el punto de vista del firewall de última generación (NGFW) de Cisco, pero los mismos conceptos también se aplican a otros tipos de dispositivos.

¿Cómo se recopilan y exportan capturas en la familia de productos NGFW?

En el caso de un appliance Firepower (1xxx, 21xx, 41xx, 93xx) y una aplicación Firepower Threat Defence (FTD), se puede visualizar el procesamiento de paquetes como se muestra en la imagen.



1. Un paquete ingresa a la interfaz de ingreso y es manejado por el switch interno del chasis.
2. El paquete entra en el motor de línea FTD que realiza principalmente comprobaciones L3/L4.
3. Si la política requiere que el paquete sea inspeccionado por el motor Snort (principalmente inspección L7).
4. El motor Snort devuelve un veredicto para el paquete.
5. El motor LINA descarta o reenvía el paquete en función del veredicto de Snort.
6. El paquete sale del chasis a través del switch de chasis interno.

Según la arquitectura mostrada, las capturas de FTD se pueden realizar en tres (3) lugares diferentes:

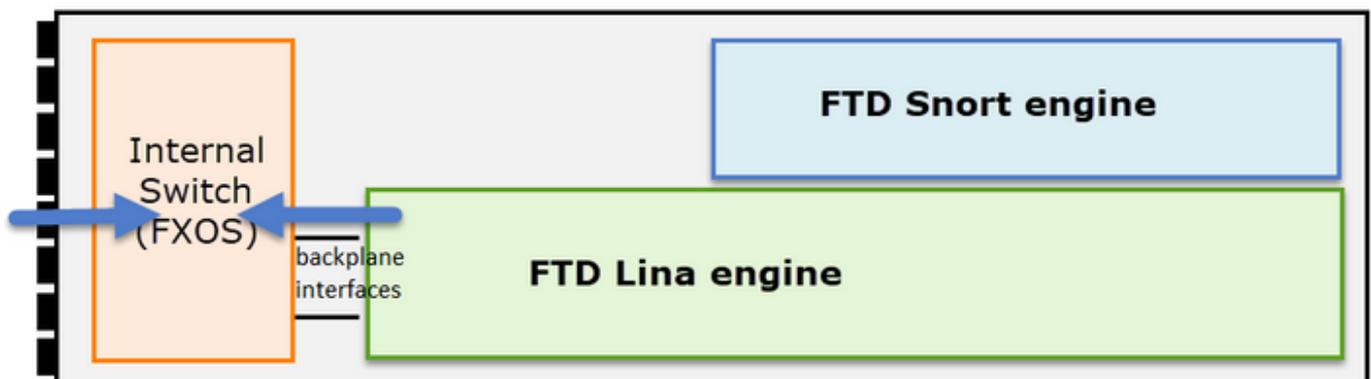
- FXOS
- Motor de línea FTD
- Motor FTD Snort

Recopilar capturas de FXOS

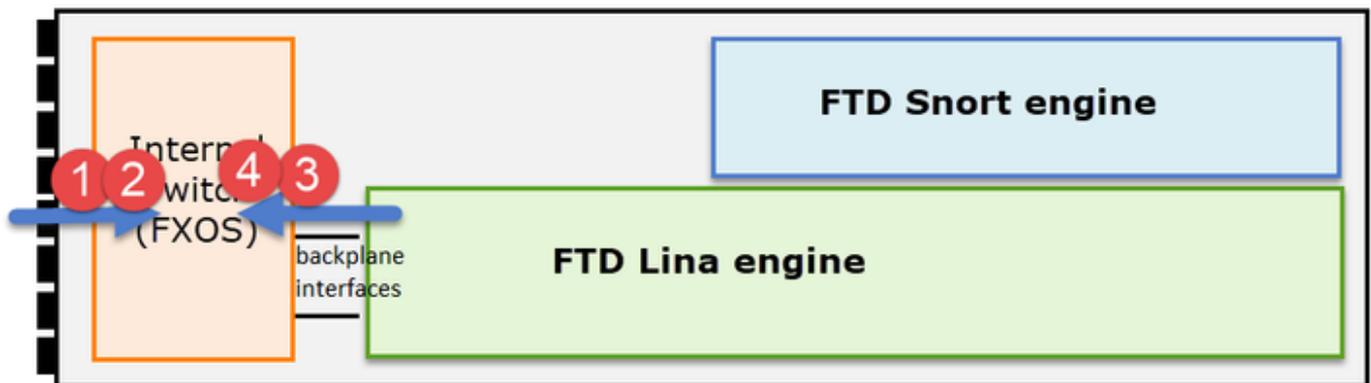
El proceso se describe en este documento:

https://www.cisco.com/c/en/us/td/docs/security/firepower/pxos/pxos271/web-guide/b_GUI_FXOS_ConfigGuide_271/troubleshooting.html#concept_E8823CC63C934A909BBC0DF12F

Las capturas de FXOS solo se pueden tomar en la dirección de ingreso desde el punto de vista del switch interno, se muestran en la imagen aquí.



Aquí se muestran dos puntos de captura por dirección (debido a la arquitectura interna del switch).



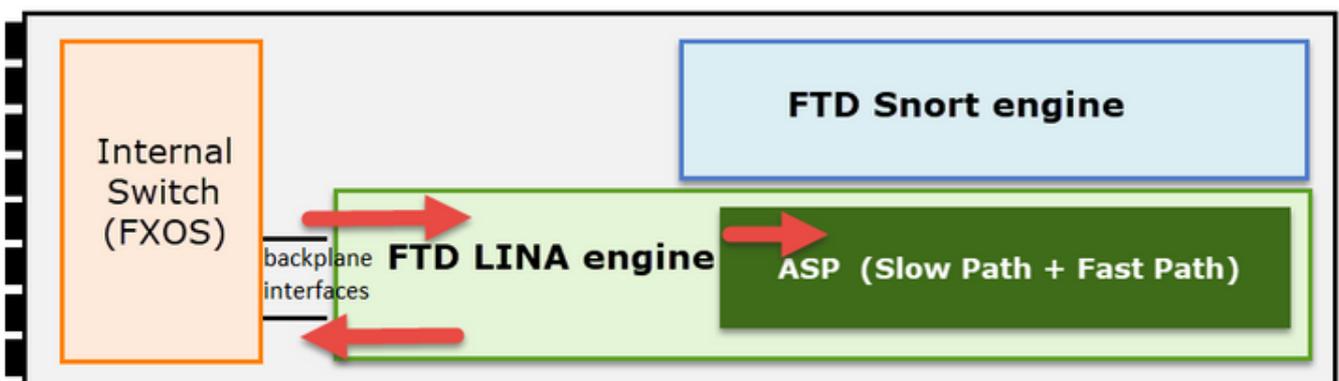
Los paquetes capturados en los puntos 2, 3 y 4 tienen una etiqueta de red virtual (VNTag).

 Nota: las capturas a nivel de chasis FXOS solo están disponibles en las plataformas FP41xx y FP93xx. FP1xxx y FP21xx no proporcionan esta capacidad.

Habilitar y recopilar capturas de línea de FTD

Puntos de captura principales:

- Interfaz de entrada
- Interfaz de salida
- Ruta de seguridad acelerada (ASP)



Puede utilizar la interfaz de usuario de Firepower Management Center (FMC UI) o la CLI de FTD para habilitar y recopilar las capturas de línea de FTD.

Habilite la captura desde CLI en la interfaz INSIDE:

```
<#root>
```

```
firepower#
```

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

Esta captura coincide con el tráfico entre las IP 192.168.103.1 y 192.168.101.1 en ambas direcciones.

Habilite la captura ASP para ver todos los paquetes descartados por el motor de línea FTD:

```
<#root>
firepower#
capture ASP type asp-drop all
```

Exportar una captura de línea de FTD a un servidor FTP:

```
<#root>
firepower#
copy /pcap capture:CAPI ftp://ftp_username:ftp_password@192.168.78.73/CAPI.pcap
```

Exportar una captura de línea FTD a un servidor TFTP:

```
<#root>
firepower#
copy /pcap capture:CAPI tftp://192.168.78.73
```

A partir de la versión FMC 6.2.x, puede habilitar y recopilar capturas de FTD Line desde la interfaz de usuario de FMC.

Otra forma de recopilar capturas de FTD de un firewall gestionado por FMC es la siguiente.

Paso 1

En el caso de una captura LINA o ASP, copie la captura en el disco FTD.

```
<#root>
firepower#
copy /pcap capture:capin disk0:capin.pcap
```

Source capture name [capin]?

Destination filename [capin.pcap]?

!!!!

Paso 2

Vaya al modo experto, localice la captura guardada y cópiela en la ubicación /ngfw/var/common:

```
<#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
```

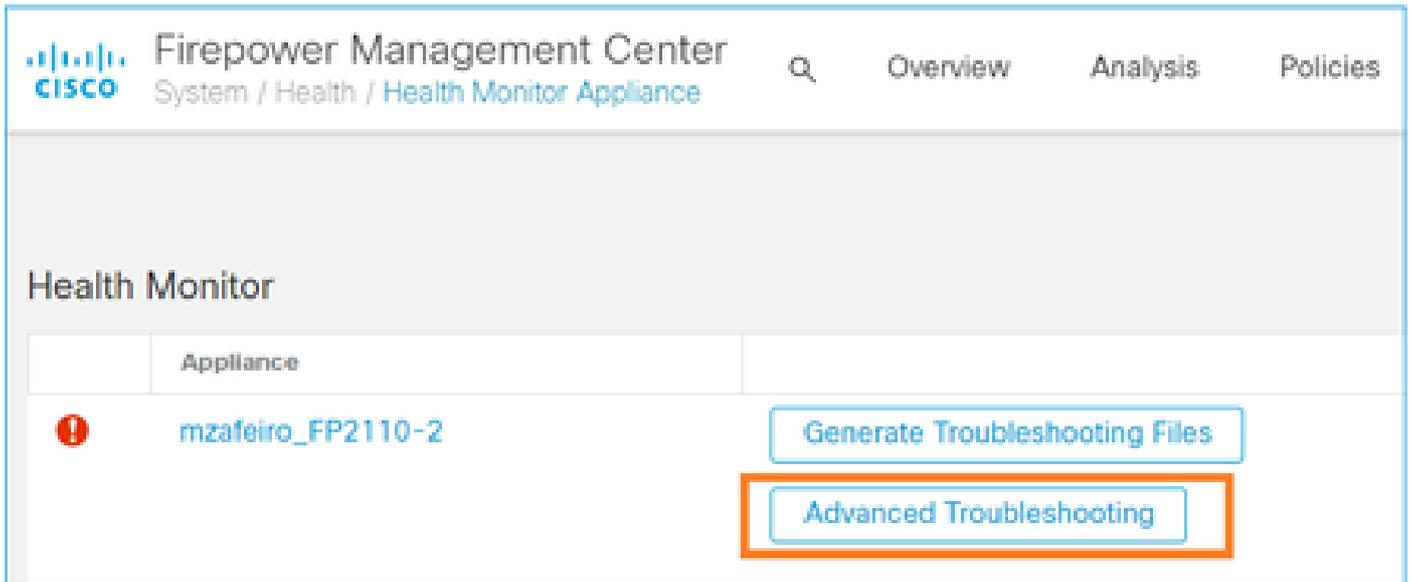
Paso 3

Inicie sesión en el FMC que gestiona el FTD y navegue hasta Dispositivos > Gestión de dispositivos. Localice el dispositivo FTD y seleccione el icono Troubleshooting:

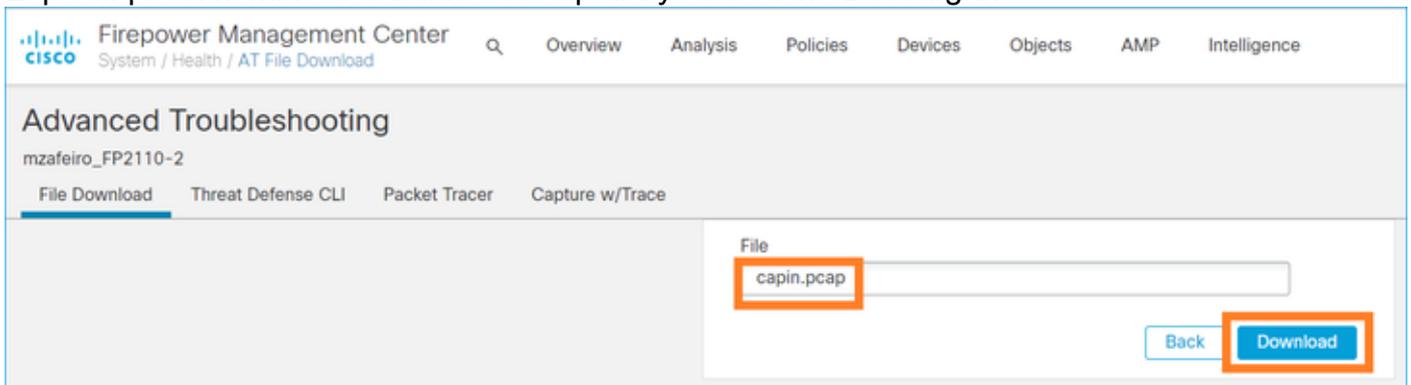


Paso 4

Seleccione Solución de problemas avanzada:



Especifique el nombre del archivo de captura y seleccione Descargar:

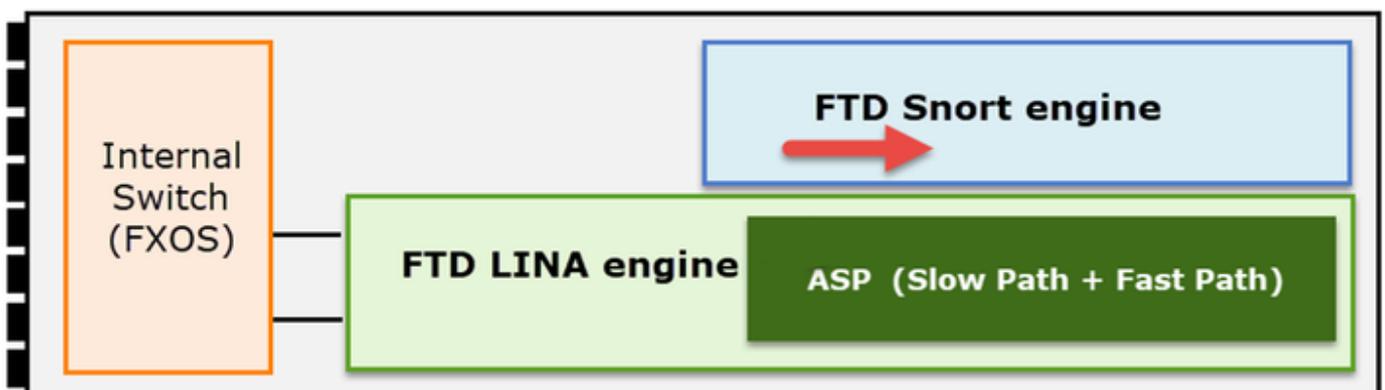


Para obtener más ejemplos sobre cómo habilitar/recopilar capturas de la interfaz de usuario de FMC, consulte este documento:

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

Activar y recopilar capturas de Snort de FTD

El punto de captura se muestra aquí en la imagen.



Habilitar captura de nivel Snort:

```
<#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
```

Para escribir la captura en un archivo con el nombre capture.pcap y copiarlo a través de FTP en un servidor remoto:

```
<#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.

>

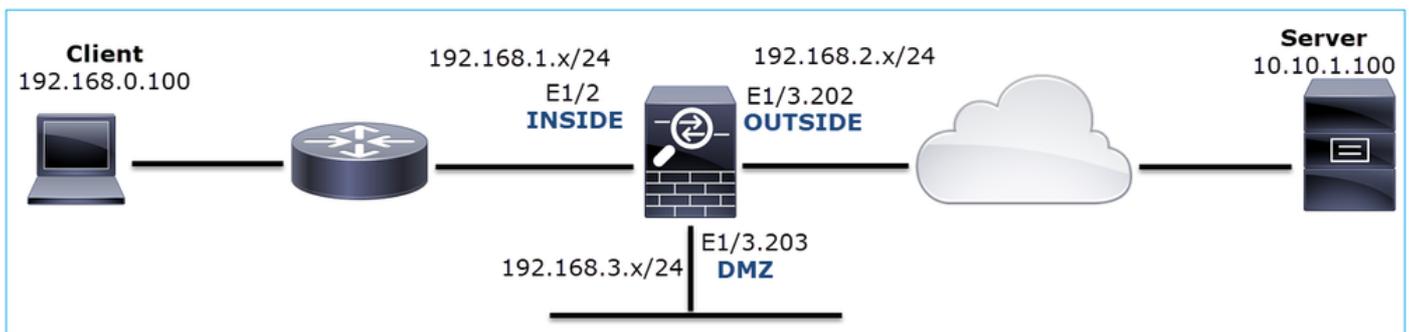
Para obtener más ejemplos de capturas de nivel Snort que incluyan diferentes filtros de captura, consulte este documento:

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

Troubleshoot

Caso 1. Sin TCP SYN en la interfaz de salida

La topología se muestra en la siguiente imagen:



Descripción del problema: HTTP no funciona

Flujo afectado:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocolo: TCP 80

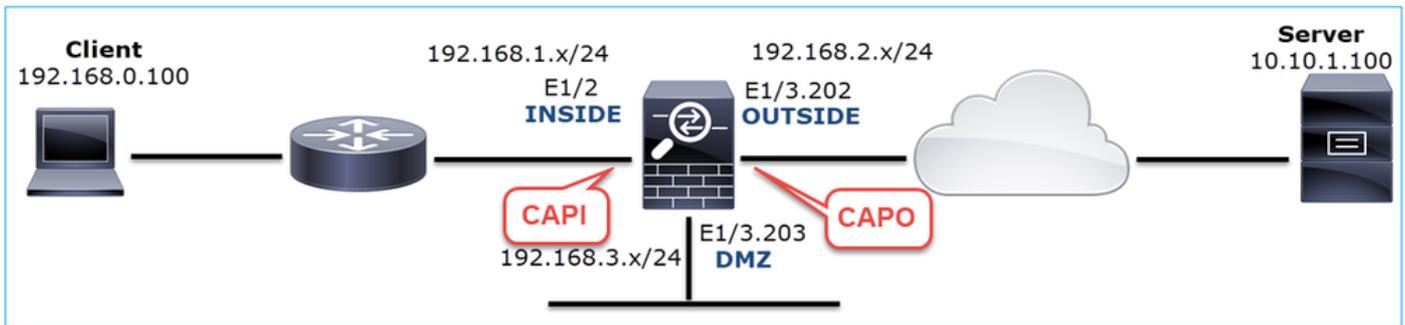
Análisis de captura

Activar capturas en el motor LINA de FTD:

```
<#root>
```

```
firepower#
```

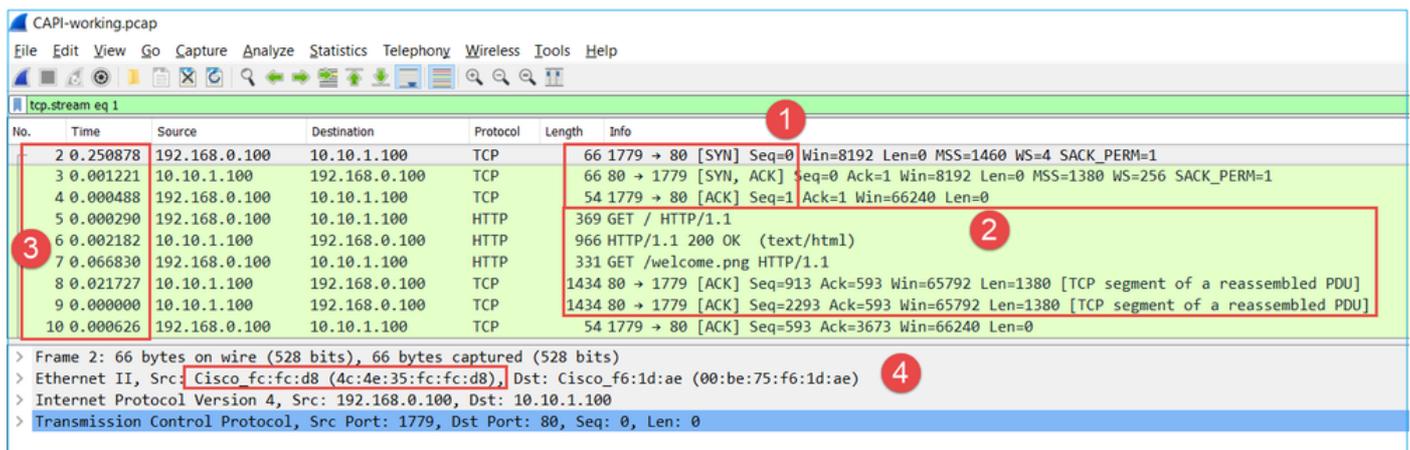
```
capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100
firepower#
capture CAPO int OUTSIDE match ip host 192.168.0.100 host 10.10.1.100
```



Capturas - Escenario funcional:

Como base, siempre es muy útil tener capturas de un escenario funcional.

La captura realizada en la interfaz NGFW INSIDE es la que se muestra en la imagen:



Puntos clave:

1. Protocolo de protocolo de enlace TCP de 3 vías.
2. Intercambio de datos bidireccional.
3. Sin retrasos entre los paquetes (según la diferencia de tiempo entre los paquetes)
4. El MAC de origen es el dispositivo de flujo descendente correcto.

La captura realizada en la interfaz exterior de NGFW se muestra en la imagen siguiente:

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: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

Puntos clave:

1. Mismos datos que en la captura CAPI.
2. El MAC de destino es el dispositivo ascendente correcto.

Capturas: escenario no funcional

Desde la CLI del dispositivo, las capturas son similares a las siguientes:

```
<#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
```

Contenido de CAPI:

```
<#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

Esta es la imagen de la captura de CAPI en Wireshark:

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	0.250470	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

3 Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

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

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

Transmission Control Protocol, Src Port: 3171, Dst Port: 80, Seq: 0, Len: 0

Puntos clave:

1. Sólo se ven los paquetes TCP SYN (sin protocolo de enlace TCP de 3 vías).
2. Hay 2 sesiones TCP (puerto de origen 3171 y 3172) que no se pueden establecer. El cliente de origen vuelve a enviar los paquetes SYN TCP. Wireshark identifica estos paquetes retransmitidos como retransmisiones TCP.

3. Las retransmisiones de TCP ocurren cada ~3 y luego cada 6 segundos, etc.
4. La dirección MAC de origen proviene del dispositivo de flujo descendente correcto.

Con base en las 2 capturas se puede concluir que:

- Un paquete de 5 tuplas específicas (IP src/dst, puerto src/dst, protocolo) llega al firewall en la interfaz esperada (INSIDE).
- Un paquete no sale del firewall en la interfaz esperada (OUTSIDE).

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Verifique el Seguimiento de un Paquete Emulado.

Utilice la herramienta packet-tracer para ver cómo se supone que el firewall debe gestionar un paquete. En caso de que la política de acceso del firewall descarte el paquete, el seguimiento del paquete emulado se verá similar a esta salida:

```
<#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

Acción 2. Verifique los seguimientos de los paquetes activos.

Habilite el seguimiento de paquetes para verificar cómo el firewall maneja los paquetes SYN TCP reales. De forma predeterminada, sólo se realiza un seguimiento de los primeros 50 paquetes de ingreso:

```
<#root>
```

```
firepower#
```

```
capture CAPI trace
```

Borre el búfer de captura:

```
<#root>
```

```
firepower#
```

```
clear capture /all
```

En caso de que la política de acceso del firewall descarte el paquete, el seguimiento será similar a este resultado:

```
<#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

Acción 3. Compruebe los registros de FTD Line.

Para configurar Syslog en FTD a través de FMC, consulte este documento:

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

Se recomienda encarecidamente tener un servidor Syslog externo configurado para los registros de FTD Line. Si no hay ningún servidor Syslog remoto configurado, habilite los registros del búfer local en el firewall mientras resuelve problemas. La configuración de registro que se muestra en este ejemplo es un buen punto inicial:

```
<#root>
firepower#
show run logging
...
logging enable
logging timestamp
logging buffer-size 1000000
logging buffered informational
```

Establezca el localizador de terminal en 24 líneas para controlar el localizador de terminal:

```
<#root>
firepower#
terminal pager 24
```

Borre el búfer de captura:

```
<#root>
firepower#
clear logging buffer
```

Pruebe la conexión y compruebe los registros con un filtro de analizador. En este ejemplo, la política de acceso del firewall descarta los paquetes:

```
<#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
```

Acción 4. Compruebe las caídas de ASP del firewall.

Si sospecha que el firewall ha descartado el paquete, puede ver los contadores de todos los paquetes descartados por el firewall a nivel de software:

```
<#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
```

Puede habilitar las capturas para ver todas las caídas de nivel de software ASP:

```
<#root>
```

```
firepower#
```

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

 Consejo: Si no está interesado en el contenido del paquete, puede capturar solamente los encabezados del paquete (opción de sólo encabezados). Esto le permite capturar muchos más paquetes en el buffer de captura. Además, puede aumentar el tamaño del búfer de captura (el valor predeterminado es 500 Kbytes) hasta un valor de hasta 32 Mbytes (opción de búfer). Por último, a partir de la versión 6.3 de FTD, la opción de tamaño de archivo permite configurar un archivo de captura de hasta 10 GBytes. En ese caso, sólo podrá ver el contenido de la captura en formato pcap.

Para comprobar el contenido de la captura, puede utilizar un filtro para restringir la búsqueda:

```
<#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
```

En este caso, dado que los paquetes ya están rastreados en el nivel de interfaz, la razón de la caída no se menciona en la captura ASP. Recuerde que un paquete sólo se puede rastrear en un lugar (interfaz de ingreso o caída de ASP). En ese caso, se recomienda tomar varias caídas de ASP y establecer un motivo de caída de ASP específico. Este es un enfoque recomendado:

1. Borre los contadores de caídas de ASP actuales:

```
<#root>
```

```
firepower#
```

```
clear asp drop
```

2. Envíe el flujo que soluciona problemas a través del firewall (realice una prueba).

3. Verifique nuevamente los contadores de caídas ASP y anote los que han aumentado.

```
<#root>
```

```
firepower#
```

```
show asp drop
```

```
Frame drop:
```

```
  No route to host (
```

```
no-route
```

```
)
```

```
234
```

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

```
acl-drop
```

```
)
```

```
71
```

4. Habilite las capturas ASP para las caídas específicas observadas:

```
<#root>
firepower#
capture ASP_NO_ROUTE type asp-drop no-route
firepower#
capture ASP_ACL_DROP type asp-drop acl-drop
```

5. Envíe el flujo que soluciona problemas a través del firewall (realice una prueba).

6. Compruebe las capturas de ASP. En este caso, los paquetes se descartaron debido a una ruta ausente:

```
<#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
```

Acción 5. Compruebe la tabla de conexiones de línea FTD.

Puede haber casos en los que se espere que el paquete salga de la interfaz 'X', pero por cualquier motivo salga de la interfaz 'Y'. La determinación de la interfaz de salida del firewall se basa en este orden de funcionamiento:

1. Búsqueda de conexión establecida
2. Búsqueda de traducción de direcciones de red (NAT): la fase UN-NAT (NAT de destino) tiene prioridad sobre PBR y la búsqueda de rutas.
3. Routing basado en políticas (PBR)
4. Búsqueda de tabla de routing

Para comprobar la tabla de conexión FTD:

```
<#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

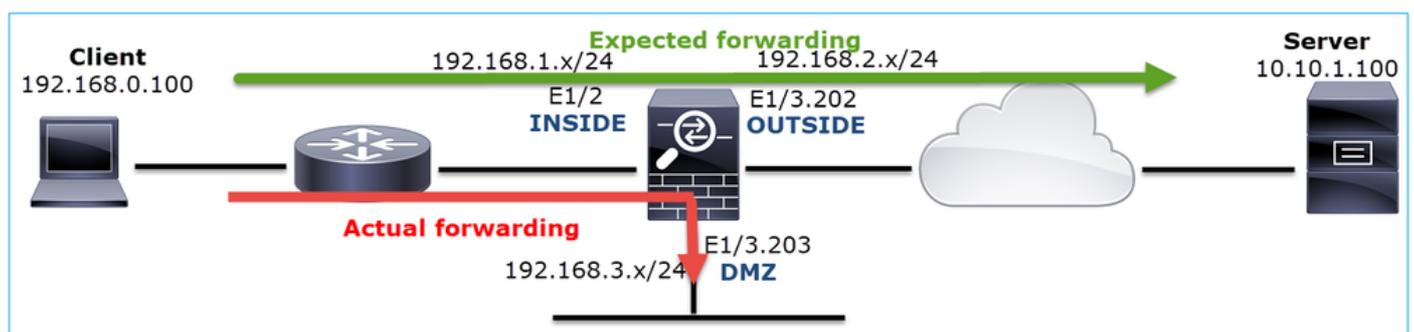
, idle 0:00:01, bytes 0, flags

aA N1

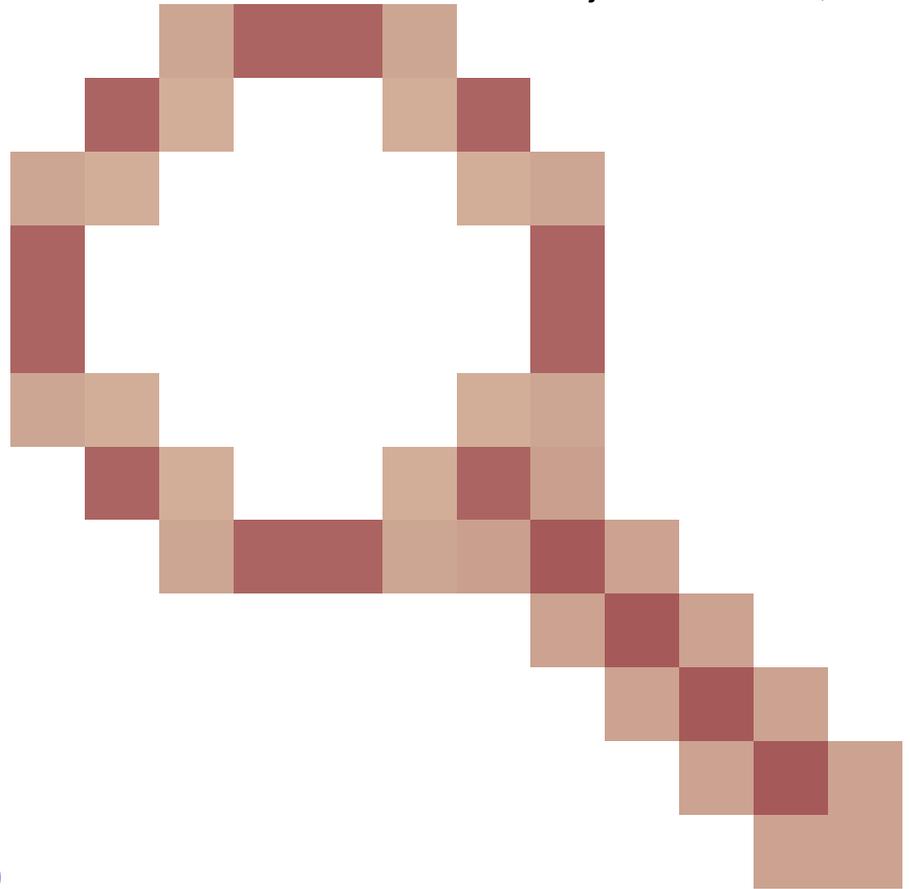
Puntos clave:

- En función de los indicadores (Aa), la conexión es embrionaria (semiabierta; el firewall solo ha visto TCP SYN).
- Según los puertos de origen/destino, la interfaz de entrada es INSIDE y la interfaz de salida es DMZ.

Esto se puede visualizar en la imagen aquí:



 Nota: Dado que todas las interfaces FTD tienen un nivel de seguridad de 0, el orden de la interfaz en la salida show conn se basa en el número de interfaz. Específicamente, la interfaz con vpif-num más alto (número de interfaz de plataforma virtual) se selecciona como interna, mientras que la interfaz con vpif-num más bajo se selecciona como externa. Puede ver el valor de la interfaz vpif con el comando show interface detail. Mejora relacionada, ID



de bug de Cisco [CSCvi15290](https://cisco.com/cisco/webbugtool/bugdetails?bugid=CSCvi15290)

ENH: FTD muestra la direccionalidad de conexión en la salida 'show conn' de FTD

```
<#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
```

 Nota: A partir de la versión 6.5 del software Firepower, la versión 9.13.x de ASA, las salidas de los comandos `show conn long` y `show conn detail` proporcionan información sobre el iniciador y el respondedor de la conexión

Resultado 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), flags
```

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

```
Connection lookup keyid: 228982375
```

Resultado 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
```

Además, el comando `show conn long` muestra las IPs NATed dentro de un paréntesis en el caso de una Traducción de Dirección de Red:

```
<#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), flags  
Initiator: 192.168.1.100, Responder: 192.168.2.222
```

```
Connection lookup keyid: 262895
```

Acción 6. Compruebe la caché del protocolo de resolución de direcciones (ARP) del firewall.

Si el firewall no puede resolver el salto siguiente, el firewall descarta silenciosamente el paquete original (TCP SYN en este caso) y envía continuamente solicitudes ARP hasta que resuelve el salto siguiente.

Para ver la memoria caché ARP del firewall, utilice el comando:

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

Además, para verificar si hay hosts sin resolver, puede utilizar el comando:

```
<#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
```

Si desea verificar más la operación ARP, puede habilitar una captura específica de ARP:

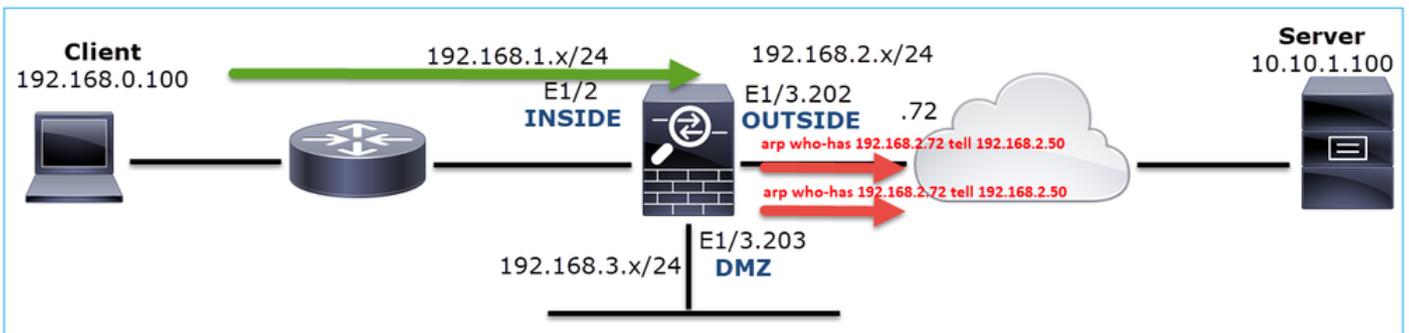
```
<#root>
firepower#
capture ARP ethernet-type arp interface OUTSIDE
```

```
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
```

En esta salida, el firewall (192.168.2.50) intenta resolver el salto siguiente (192.168.2.72), pero no hay respuesta ARP



El resultado aquí muestra un escenario funcional con una resolución ARP adecuada:

```
<#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
```

En caso de que no haya ninguna entrada ARP en el lugar, un seguimiento de un paquete SYN TCP activo muestra:

```
<#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
```

Como se puede ver en el resultado, el seguimiento muestra Action: allow incluso cuando el salto siguiente no es alcanzable y el paquete es silenciosamente descartado por el firewall. En este caso, la herramienta packet-tracer también debe ser verificada ya que proporciona una salida más precisa:

```
<#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),

En las últimas versiones de ASA/Firepower, el mensaje anterior se ha optimizado para:

<#root>

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

Check ARP table (show arp) has entry for nexthop

., Drop-location: f

Resumen de posibles causas y acciones recomendadas

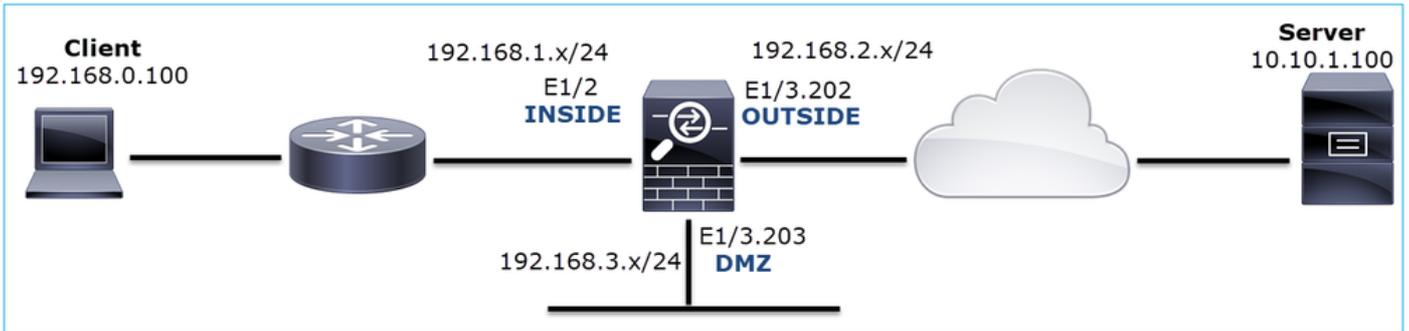
Si sólo ve un paquete TCP SYN en las interfaces de ingreso, pero ningún paquete TCP SYN enviado desde la interfaz de egreso esperada, algunas causas posibles son:

Posible Causa	Acciones recomendadas
La política de acceso del firewall descarta el paquete.	<ul style="list-style-type: none">• Utilice packet-tracer o capture w/trace para ver cómo el firewall maneja el paquete.• Compruebe los registros del firewall.• Verifique las caídas de ASP del firewall (show asp drop o capture type asp-drop).• Compruebe los eventos de conexión FMC. Esto supone que la regla tiene el registro habilitado.
El filtro de captura es incorrecto.	<ul style="list-style-type: none">• Utilice packet-tracer o capture w/trace para ver si hay traducción NAT que modifique la IP de origen o de destino. En ese caso, ajuste el filtro de captura.• El resultado del comando show conn long muestra las IPs NATed.

<p>El paquete se envía a una interfaz de salida diferente.</p>	<ul style="list-style-type: none"> • Utilice packet-tracer o capture w/trace para ver cómo el firewall maneja el paquete. Recuerde el orden de las operaciones que se refieren a la determinación de la interfaz de egreso, la conexión actual, UN-NAT, PBR y la búsqueda de la tabla de ruteo. • Compruebe los registros del firewall. • Verifique la tabla de conexión del firewall (show conn). <p>Si el paquete se envía a una interfaz incorrecta porque coincide con una conexión actual, utilice el comando clear conn address y especifique la 5-tupla de la conexión que desea borrar.</p>
<p>No hay ruta hacia el destino.</p>	<ul style="list-style-type: none"> • Utilice packet-tracer o capture w/trace para ver cómo el firewall maneja el paquete. • Verifique las caídas de ASP del firewall (show asp drop) para ver el motivo de la caída sin ruta.
<p>No hay ninguna entrada ARP en la interfaz de salida.</p>	<ul style="list-style-type: none"> • Verifique la memoria caché ARP del firewall (show arp). • Utilice packet-tracer para ver si hay una adyacencia válida.
<p>La interfaz de salida está inactiva.</p>	<p>Verifique la salida del comando show interface ip brief en el firewall y verifique el estado de la interfaz.</p>

Caso 2. TCP SYN del cliente, TCP RST del servidor

Esta imagen muestra la topología:



Descripción del problema: HTTP no funciona

Flujo afectado:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocolo: TCP 80

Análisis de captura

Activar capturas en el motor LINA de FTD.

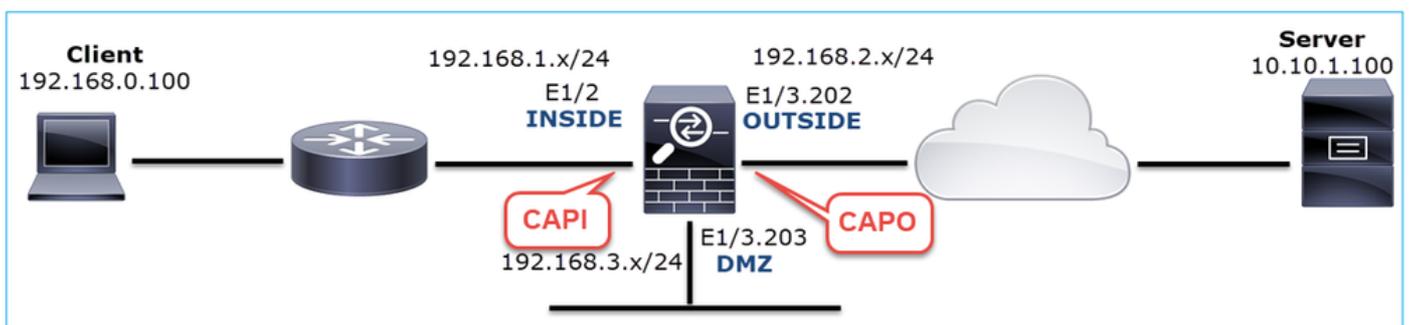
```
<#root>
```

```
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
```



Capturas - Escenario no funcional:

Desde la CLI del dispositivo, las capturas son las siguientes:

```
<#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
```

Contenido de CAPI:

```
<#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>
```

```
...
```

Contenido de CAPO:

```
<#root>
```

firepower#

show capture CAPO

```
1: 05:20:36.654507 802.1Q vlan#202 PO 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 PO 192.168.0.100.22196 > 10.10.1.100.80:
S
4785344:4785344(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>
3: 05:20:36.904997 802.1Q vlan#202 PO 10.10.1.100.80 > 192.168.0.100.22196:
R
0:0(0) ack 4785345 win 0
4: 05:20:37.414269 802.1Q vlan#202 PO 192.168.0.100.22196 > 10.10.1.100.80:
S
4235354730:4235354730(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>
5: 05:20:37.414758 802.1Q vlan#202 PO 10.10.1.100.80 > 192.168.0.100.22196:
R
0:0(0) ack 4235354731 win 0
6: 05:20:37.914305 802.1Q vlan#202 PO 192.168.0.100.22196 > 10.10.1.100.80:
S
4118617832:4118617832(0) win 8192 <mss 1380,nop,nop,sackOK>
```

Esta imagen muestra la captura de CAPI en Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.100	10.10.1.100	TCP	66	22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.250094	192.168.0.100	10.10.1.100	TCP	66	22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	0.000732	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	0.509089	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
5	0.000671	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=2476911971 Ack=1 Win=0 Len=0
6	0.499380	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
7	0.000625	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=2853655305 Ack=1 Win=0 Len=0
8	1.739729	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
9	0.000611	10.10.1.100	192.168.0.100	TCP	54	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10	0.499385	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
11	0.000671	10.10.1.100	192.168.0.100	TCP	54	80 → 22195 [RST, ACK] Seq=151733665 Ack=1 Win=0 Len=0

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc: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

Puntos clave:

1. El origen envía un paquete TCP SYN.
2. Se envía un TCP RST hacia el origen.
3. El origen retransmite los paquetes TCP SYN.
4. Las direcciones MAC son correctas (en los paquetes de entrada, la dirección MAC de origen pertenece al router de flujo descendente, la dirección MAC de destino pertenece a la interfaz interna del firewall).

Esta imagen muestra la captura de CAPO en Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-11 07:20:36.654507	192.168.0.100	10.10.1.100	TCP	70	22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
2	2019-10-11 07:20:36.904478	192.168.0.100	10.10.1.100	TCP	70	22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
3	2019-10-11 07:20:36.904997	10.10.1.100	192.168.0.100	TCP	58	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	2019-10-11 07:20:37.414269	192.168.0.100	10.10.1.100	TCP	70	[TCP Port numbers reused] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
5	2019-10-11 07:20:37.414758	10.10.1.100	192.168.0.100	TCP	58	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6	2019-10-11 07:20:37.914305	192.168.0.100	10.10.1.100	TCP	66	[TCP Port numbers reused] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 SACK_PERM=1
7	2019-10-11 07:20:37.914762	10.10.1.100	192.168.0.100	TCP	58	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
8	2019-10-11 07:20:39.654629	192.168.0.100	10.10.1.100	TCP	70	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
9	2019-10-11 07:20:39.655102	10.10.1.100	192.168.0.100	TCP	58	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10	2019-10-11 07:20:40.154700	192.168.0.100	10.10.1.100	TCP	66	[TCP Port numbers reused] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 SACK_PERM=1
11	2019-10-11 07:20:40.155173	10.10.1.100	192.168.0.100	TCP	58	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

<

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Cisco_fc: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

Puntos clave:

1. El origen envía un paquete TCP SYN.
2. Llega un TCP RST a la interfaz OUTSIDE.
3. El origen retransmite los paquetes TCP SYN.
4. Las direcciones MAC son correctas (en los paquetes de salida, el firewall OUTSIDE es el MAC de origen, el router ascendente es el MAC de destino).

Con base en las 2 capturas se puede concluir que:

- El intercambio de señales TCP de 3 vías entre el cliente y el servidor no se completa
- Hay un TCP RST que llega a la interfaz de salida del firewall
- El firewall "habla" con los dispositivos de flujo ascendente y descendente adecuados (según las direcciones MAC)

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Verifique la dirección MAC de origen que envía el TCP RST.

Verifique que el MAC de destino visto en el paquete TCP SYN sea el mismo que el MAC de origen visto en el paquete TCP RST.

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, Prio: 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, Prio: 0, DEI: 0, ID: 202

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

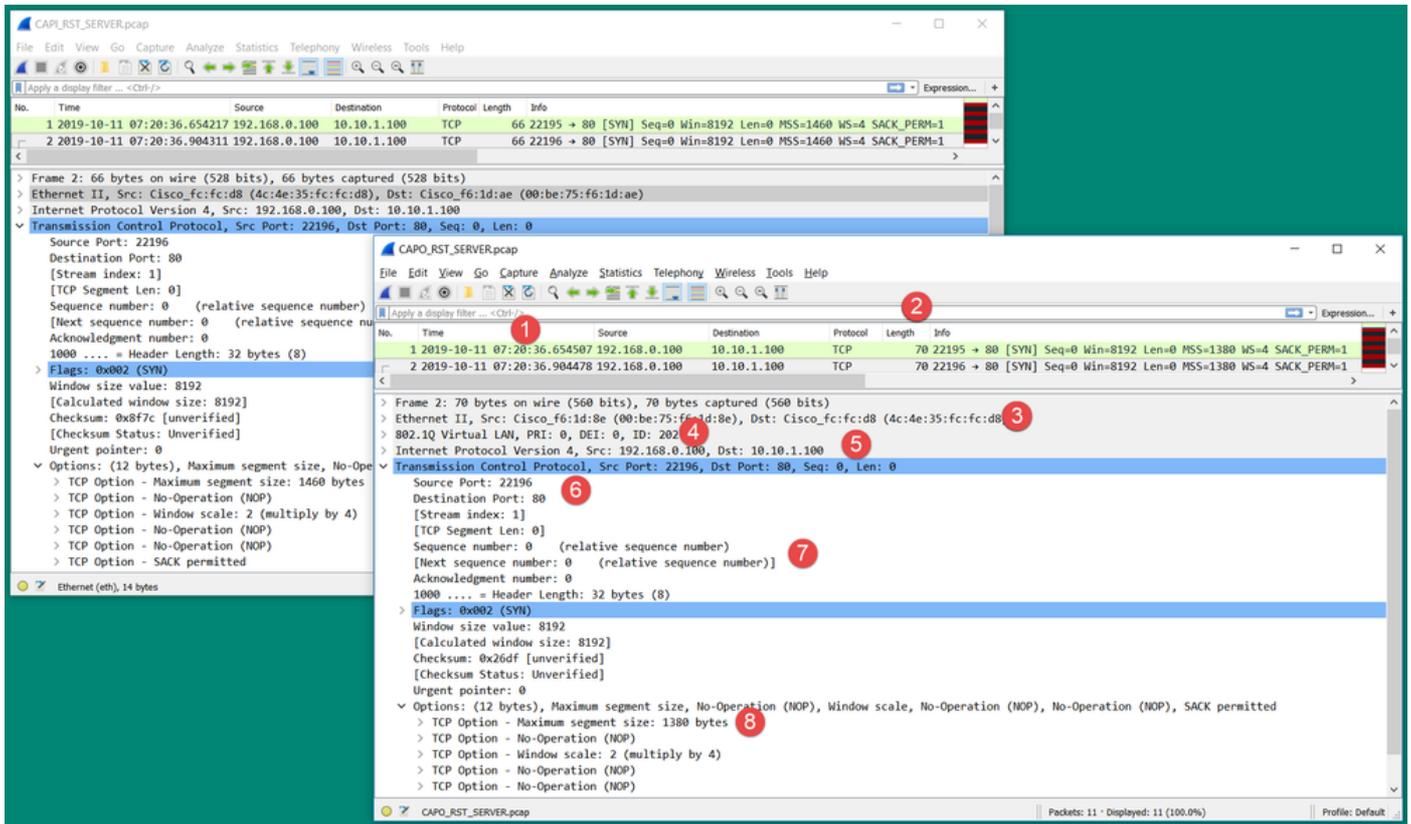
> Transmission Control Protocol, Src Port: 80, Dst Port: 22196, Seq: 1, Ack: 1, Len: 0

Esta comprobación tiene como objetivo confirmar 2 cosas:

- Verifique que no haya ningún flujo asimétrico.
- Verifique que el MAC pertenezca al dispositivo ascendente esperado.

Acción 2. Compare los paquetes de entrada y salida.

Compare visualmente los 2 paquetes de Wireshark para verificar que el firewall no modifique/dañe los paquetes. Se resaltan algunas diferencias esperadas.



Puntos clave:

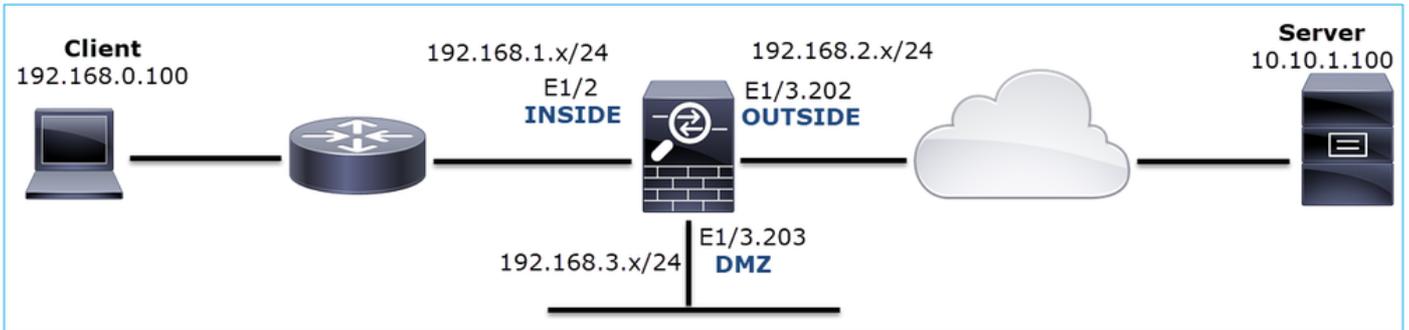
1. Las marcas de tiempo son diferentes. Por otro lado, la diferencia debe ser pequeña y razonable. Esto depende de las funciones y las comprobaciones de políticas aplicadas al paquete, así como de la carga en el dispositivo.
2. La longitud de los paquetes puede diferir especialmente si hay un encabezado dot1Q agregado/eliminado por el firewall en un solo lado.
3. Las direcciones MAC son diferentes.
4. Un encabezado dot1Q puede estar en su lugar si la captura fue tomada en una subinterfaz.
5. Las direcciones IP son diferentes en caso de que se aplique NAT o traducción de direcciones de puerto (PAT) al paquete.
6. Los puertos de origen o de destino son diferentes en caso de que se aplique NAT o PAT al paquete.
7. Si inhabilita la opción Wireshark Relative Sequence Number, verá que los números de secuencia TCP/números de reconocimiento son modificados por el firewall debido a la aleatorización del Número de secuencia inicial (ISN).
8. Algunas opciones TCP se pueden sobrescribir. Por ejemplo, el firewall cambia de forma predeterminada el tamaño máximo de segmento (MSS) de TCP a 1380 para evitar la fragmentación de paquetes en la ruta de tránsito.

Acción 3. Toma una captura en el destino.

Si es posible, tome una captura en el propio destino. Si esto no es posible, tome una captura lo más cerca posible del destino. El objetivo aquí es verificar quién envía el TCP RST (¿es el servidor de destino o hay algún otro dispositivo en la trayectoria?).

Caso 3. Protocolo de enlace TCP de 3 vías + RST desde un terminal

Esta imagen muestra la topología:



Descripción del problema: HTTP no funciona

Flujo afectado:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocolo: TCP 80

Análisis de captura

Activar capturas en el motor LINA de FTD.

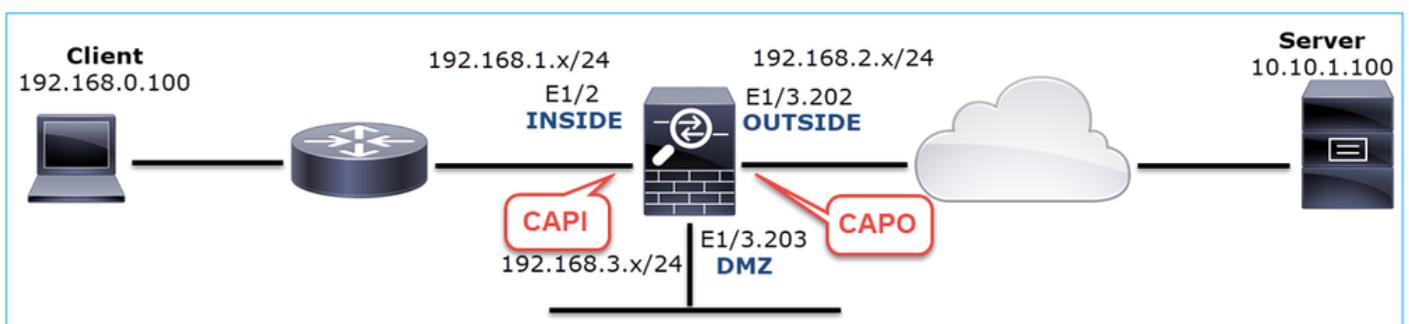
<#root>

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
```



Capturas - Escenario no funcional:

Hay un par de maneras diferentes en que este problema puede manifestarse en capturas.

3.1 - Protocolo de enlace TCP de 3 vías + RST retrasado del cliente

Tanto el firewall captura CAPI como CAPO contienen los mismos paquetes, como se muestra en la imagen.

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

Puntos clave:

1. El protocolo de enlace de 3 vías TCP pasa a través del firewall.
2. El servidor retransmite el SYN/ACK.
3. El cliente retransmite el ACK.
4. Después de ~20 segundos, el cliente se da por vencido y envía un TCP RST.

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Realice las capturas lo más cerca posible de los dos terminales.

Las capturas del firewall indican que el servidor no procesó el ACK del cliente. Esto se basa en los siguientes hechos:

- El servidor retransmite el SYN/ACK.
- El cliente retransmite el ACK.
- El cliente envía un TCP RST o FIN/ACK antes de cualquier dato.

La captura en el servidor muestra el problema. El ACK del cliente del intercambio de señales TCP de 3 vías nunca llegó:

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 - Protocolo de enlace TCP de 3 vías + FIN/ACK retrasado del cliente + RST retrasado del servidor

Tanto el firewall captura CAPI como CAPO contienen los mismos paquetes, como se muestra en

la imagen.

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

Puntos clave:

1. El protocolo de enlace de 3 vías TCP pasa a través del firewall.
2. Después de unos 5 segundos, el cliente envía un mensaje FIN/ACK.
3. Después de unos 20 segundos, el servidor se rinde y envía un TCP RST.

En base a esta captura, se puede concluir que aunque existe un protocolo de enlace TCP de 3 vías a través del firewall, parece que nunca se completa realmente en un terminal (las retransmisiones indican esto).

Acciones recomendadas

Igual que en el caso 3.1

3.3 - Protocolo de enlace TCP de 3 vías + RST retrasado del cliente

Tanto el firewall captura CAPI como CAPO contienen los mismos paquetes, como se muestra en la imagen.

No.	Time	Source	Destination	Protocol	Length	Info	
129	2019-10-13	17:09:20.513355	192.168.0.100	10.10.1.100	TCP	66	48355 → 80 [SYN] Seq=2581697538 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
130	2019-10-13	17:09:20.514011	10.10.1.100	192.168.0.100	TCP	66	80 → 48355 [SYN, ACK] Seq=1633018698 Ack=2581697539 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
131	2019-10-13	17:09:20.514438	192.168.0.100	10.10.1.100	TCP	54	48355 → 80 [ACK] Seq=2581697539 Ack=1633018699 Win=66240 Len=0
132	2019-10-13	17:09:39.473089	192.168.0.100	10.10.1.100	TCP	54	80 → 48355 [RST, ACK] Seq=2581697939 Ack=1633018699 Win=0 Len=0

Puntos clave:

1. El protocolo de enlace de 3 vías TCP pasa a través del firewall.
2. Después de ~20 segundos, el cliente se da por vencido y envía un TCP RST.

Sobre la base de estas capturas, se puede concluir que:

- Después de 5-20 segundos, un terminal se da por vencido y decide terminar la conexión.

Acciones recomendadas

Igual que en el caso 3.1

3.4 - Protocolo de enlace TCP de 3 vías + RST inmediato desde el servidor

Tanto las capturas de firewall CAPI como CAPO contienen estos paquetes, como se muestra en

la imagen.

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

Puntos clave:

1. El protocolo de enlace de 3 vías TCP pasa a través del firewall.
2. Hay un TCP RST del servidor unos milisegundos después del paquete ACK.

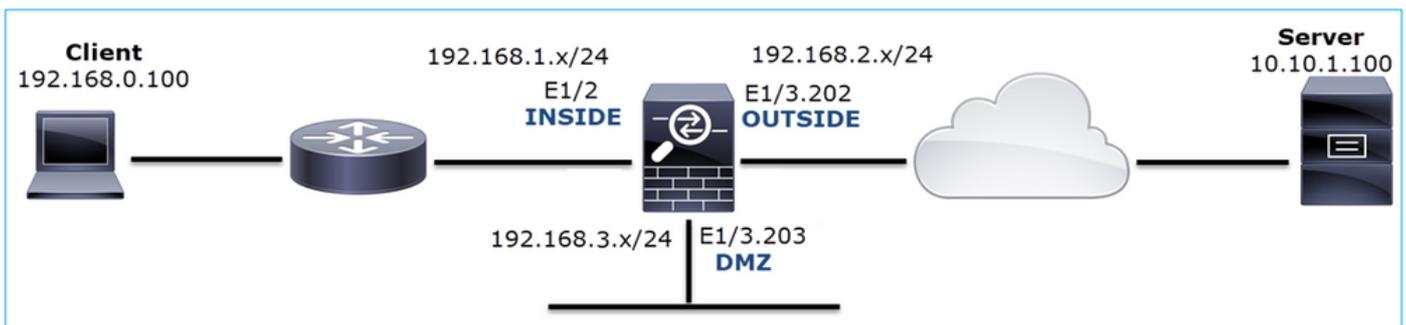
Acciones recomendadas

Acción: realice las capturas lo más cerca posible del servidor.

Un TCP RST inmediato del servidor podría indicar un servidor que no funciona correctamente o un dispositivo en la trayectoria que envía el TCP RST. Realice una captura en el propio servidor y determine el origen del TCP RST.

Caso 4. TCP RST desde el cliente

Esta imagen muestra la topología:



Descripción del problema: HTTP no funciona.

Flujo afectado:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocolo: TCP 80

Análisis de captura

Activar capturas en el motor LINA de FTD.

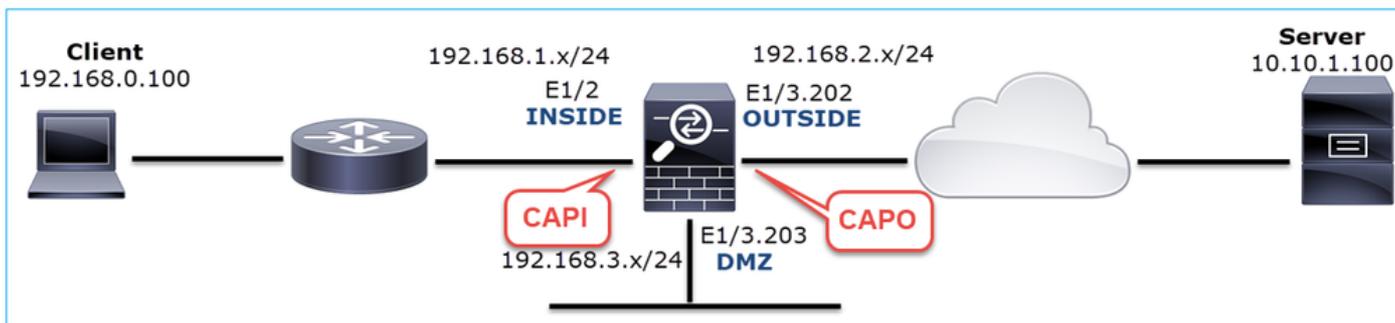
```
<#root>
```

```
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
```



Capturas - Escenario no funcional:

Estos son los contenidos de CAPI.

```
<#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

Estos son los contenidos de CAPO:

```
<#root>
```

```
firepower#
```

```
show capture CAPO
```

```
11 packets captured
```

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

```
11 packets shown
```

Los registros del firewall muestran:

```
<#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
```

Estos registros indican que hay un TCP RST que llega a la interfaz de firewall INSIDE

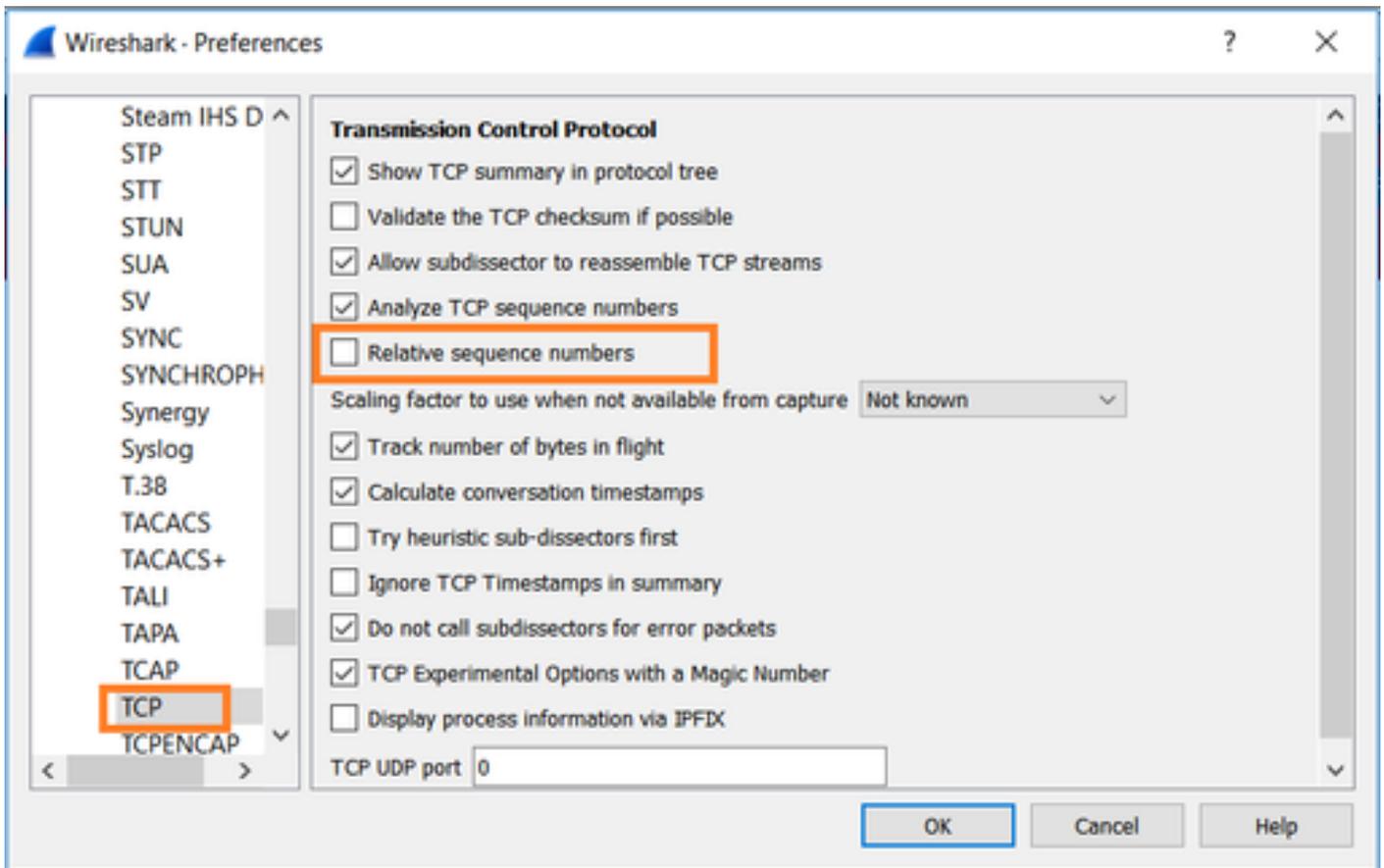
Captura de CAPI en Wireshark:

Siga la primera secuencia TCP, como se muestra en la imagen.

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_P...
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_P...
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

- Mark/Unmark Packet
- Ignore/Unignore Packet
- Set/Unset Time Reference
- Time Shift...
- Packet Comment...
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow
 - TCP Stream
 - UDP Stream
 - SSL Stream
 - HTTP Stream
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

En Wireshark, navegue hasta Edit > Preferences > Protocols > TCP y deseleccione la opción Relative sequence numbers como se muestra en la imagen.



Esta imagen muestra el contenido del primer flujo en la captura CAPI:

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
> 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]

```

Puntos clave:

1. El cliente envía un paquete TCP SYN.
2. El cliente envía un paquete RST TCP.
3. El paquete TCP SYN tiene un valor de número de secuencia igual a 4098574664.

El mismo flujo en la captura de CAPO contiene:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-13 14:32:22.860780	192.168.0.100	10.10.1.100	TCP	70	47078 → 80 [SYN] Seq=1386249852 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
4	2019-10-13 14:32:25.858125	192.168.0.100	10.10.1.100	TCP	70	[TCP Retransmission] 47078 → 80 [SYN] Seq=1386249852 Win=8192 Len=0 MSS=1380
5	2019-10-13 14:32:25.868729	192.168.0.100	10.10.1.100	TCP	58	47078 → 80 [RST] Seq=2968892337 Win=0 Len=0


```

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
> Ethernet II, Src: Cisco_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: 47078, Dst Port: 80, Seq: 1386249852, Len: 0

```

Puntos clave:

1. El cliente envía un paquete TCP SYN. El firewall aleatoriza el ISN.
2. El cliente envía un paquete RST TCP.

Con base en las dos capturas se puede concluir que:

- No hay intercambio de señales TCP de 3 vías entre el cliente y el servidor.
- Hay un TCP RST que viene del cliente. El valor del número de secuencia RST TCP en la captura CAPI es 1386249853.

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Tome una captura en el cliente.

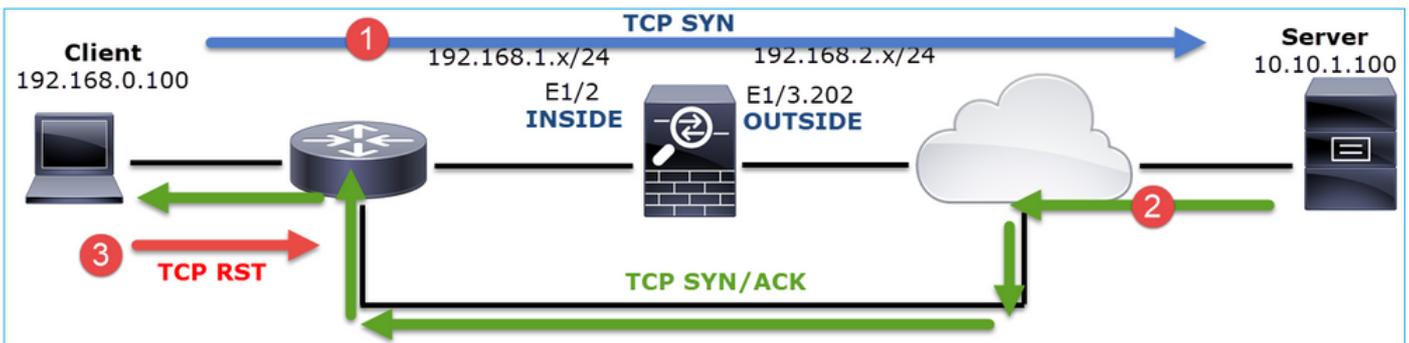
En función de las capturas recopiladas en el firewall, existe un fuerte indicador de un flujo asimétrico. Esto se basa en el hecho de que el cliente envía un TCP RST con un valor de 1386249853 (el ISN aleatorizado):

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 1 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 3

Puntos clave:

1. El cliente envía un paquete TCP SYN. El número de secuencia es 4098574664 y es el mismo que se ve en la interfaz interna del firewall (CAPI)
2. Hay un TCP SYN/ACK con el número ACK 1386249853 (que se espera debido a la aleatorización ISN). Este paquete no se vio en las capturas del firewall
3. El cliente envía un TCP RST ya que esperaba un SYN/ACK con un valor de número ACK de 4098574665, pero recibió un valor de 1386249853

Esto se puede visualizar como:

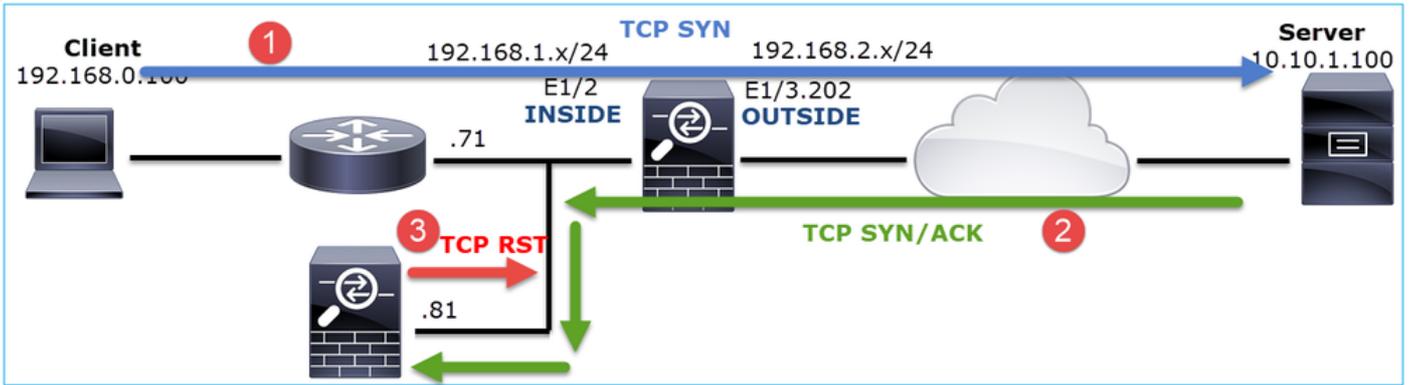


Acción 2. Compruebe el enrutamiento entre el cliente y el firewall.

Confirme que:

- Las direcciones MAC vistas en las capturas son las esperadas.
- Asegúrese de que el routing entre el firewall y el cliente sea simétrico.

Hay escenarios donde el RST proviene de un dispositivo que se encuentra entre el firewall y el cliente mientras hay un ruteo asimétrico en la red interna. En la imagen se muestra un caso típico:



En este caso, la captura tiene este contenido. Observe la diferencia entre la dirección MAC de origen del paquete TCP SYN y la dirección MAC de origen del RST TCP y la dirección MAC de destino del paquete TCP SYN/ACK:

```
<#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)
```

```
...
```

Caso 5. Transferencia TCP lenta (situación 1)

Descripción de problemas:

La transferencia SFTP entre los hosts 10.11.4.171 y 10.77.19.11 es lenta. Aunque el ancho de banda mínimo (BW) entre los 2 hosts es de 100 Mbps, la velocidad de transferencia no supera los 5 Mbps.

Al mismo tiempo, la velocidad de transferencia entre los hosts 10.11.2.124 y 172.25.18.134 es bastante mayor.

Teoría Precedente:

La velocidad máxima de transferencia para un solo flujo TCP está determinada por el producto de retraso de ancho de banda (BDP). La fórmula utilizada se muestra en la imagen:

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

Para obtener más información sobre la BDP, consulte los recursos aquí:

- [¿Por qué su aplicación sólo utiliza 10 Mbps incluso si el enlace es de 1 Gbps?](#)
- [BRKSEC-3021 - Avanzado - Maximización del rendimiento del firewall](#)

Escenario 1. Transferencia lenta

Esta imagen muestra la topología:



Flujo afectado:

Src IP: 10.11.4.171

Dst IP: 10.77.19.11

Protocolo: SFTP (FTP sobre SSH)

Análisis de captura

Habilitar capturas en el motor LINA de FTD:

<#root>

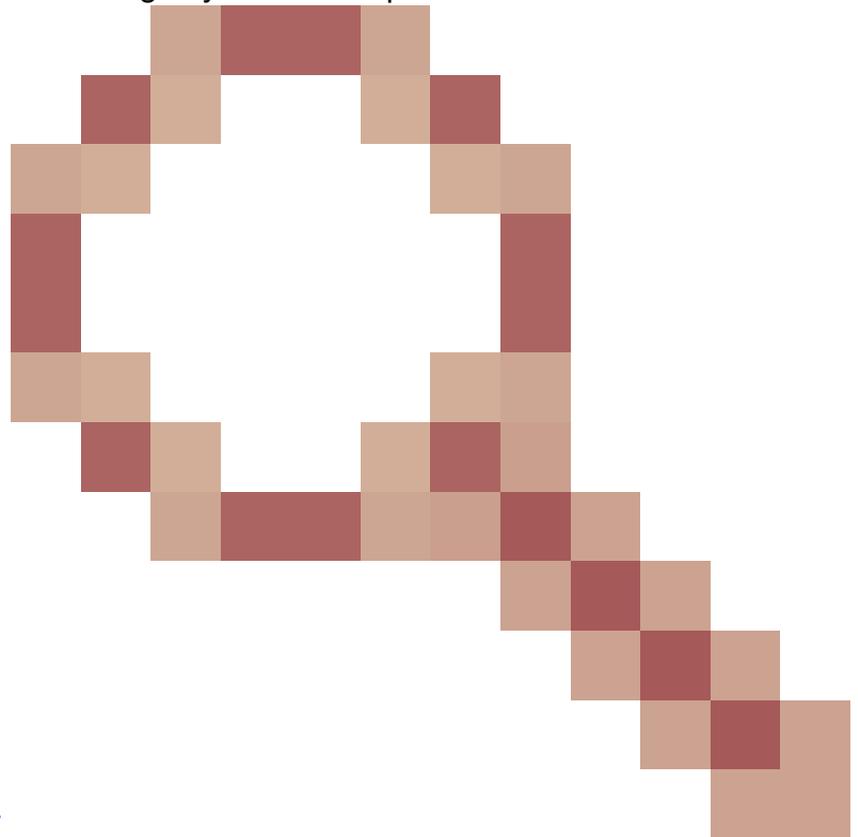
```
firepower#
```

```
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
```

 Advertencia: las capturas de LINA en FP1xxx y FP21xx afectan a la velocidad de transferencia del tráfico que pasa a través del FTD. No habilite las capturas de LINA en las plataformas FP1xxx y FP21xxx cuando resuelva problemas de rendimiento (transferencia lenta a través del FTD). En su lugar, utilice SPAN o un dispositivo de toque de hardware además de las capturas en los hosts de origen y destino. El problema se documenta con el



ID de bug de Cisco [CSCvo30697](https://www.cisco.com/cisco/webbugtool/bugdetails.do?bugid=CSCvo30697)

```
<#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.
```

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

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 ms

Cálculo del Tamaño de Ventana TCP

Expanda un paquete TCP, expanda el encabezado TCP, seleccione Tamaño de ventana calculado y seleccione Aplicar como columna:

Transmission Control Protocol, Src Port: 22, Dst Port: 39744, Seq: 835184024, Ack: 1758069308, Len: 32

- Source Port: 22
- Destination Port: 39744
- [Stream index: 0]
- [TCP Segment Len: 32]
- Sequence number: 835184024
- [Next sequence number: 835184056]
- Acknowledgment number: 1758069308
- 0101 = Header Length: 20 bytes (5)
- > Flags: 0x018 (PSH, ACK)
- Window size value: 49680
- [Calculated window size: 49680]
- [Window size scaling factor: ...]
- Checksum: 0x2b49 [unverified]
- [Checksum Status: Unverified]

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

Verifique la columna Valor de tamaño de ventana calculado para ver cuál fue el valor de tamaño de ventana máximo durante la sesión TCP. También puede seleccionar el nombre de la columna y ordenar los valores.

Si prueba una descarga de archivos (servidor > cliente), debe comprobar los valores anunciados por el servidor. El valor del tamaño máximo de la ventana anunciado por el servidor determina la velocidad máxima de transferencia alcanzada.

En este caso, el tamaño de la ventana TCP es ≈ 50000 bytes

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)

En función de estos valores y con el uso de la fórmula Bandwidth Delay Product, se obtiene el ancho de banda teórico máximo que se puede alcanzar en estas condiciones: $50000 * 8/0,08 =$

ancho de banda teórico máximo de 5 Mbps.

Esto coincide con lo que el cliente experimenta en este caso.

Verifique atentamente el protocolo de enlace TCP de 3 vías. Ambos lados, y más importante aún el servidor, anuncian un valor de escala de ventana de 0 que significa $2^0 = 1$ (sin escala de ventanas). Esto afecta negativamente a la velocidad de transferencia:

```
No. Time Source Destination Protocol Length Window size value Info
1 0.000000 10.11.4.171 10.77.19.11 TCP 70 49640 39744 → 22 [SYN] Seq=1737026093 Win=49640 Len=0 MSS=1460 WS=1 SACK_PERM=1
2 0.072521 10.77.19.11 10.11.4.171 TCP 70 49680 22 → 39744 [SYN, ACK] Seq=835172681 Ack=1737026094 Win=49680 Len=0 MSS=1380 WS=1 SACK_PERM=1
<
> Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
> Ethernet II, Src: Cisco_1f:72:4e (00:5d:73:1f:72:4e), Dst: Cisco_f8:19:ff (00:22:bd:f8:19:ff)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
> Internet Protocol Version 4, Src: 10.77.19.11, Dst: 10.11.4.171
> Transmission Control Protocol, Src Port: 22, Dst Port: 39744, Seq: 835172681, Ack: 1737026094, Len: 0
  Source Port: 22
  Destination Port: 39744
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 835172681
  [Next sequence number: 835172681]
  Acknowledgment number: 1737026094
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x012 (SYN, ACK)
  Window size value: 49680
  [Calculated window size: 49680]
  Checksum: 0xa91b [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
    > TCP Option - Maximum segment size: 1380 bytes
    > TCP Option - No-Operation (NOP)
    > TCP Option - Window scale: 0 (multiply by 1)
    > TCP Option - No-Operation (NOP)
```

En este punto, hay una necesidad de tomar una captura en el servidor, confirmar que es el que anuncia la escala de ventana = 0 y reconfigurarlo (consulte la documentación del servidor para ver cómo hacer esto).

Situación hipotética 2. Transferencia rápida

Ahora examinemos el buen escenario (transferencia rápida a través de la misma red):

Topología:



El flujo de interés:

Src IP: 10.11.2.124

Dst IP: 172.25.18.134

Protocolo: SFTP (FTP sobre SSH)

Activar capturas en el motor LINA de FTD

<#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
```

Cálculo del tiempo de ida y vuelta (RTT): en este caso, el RTT es ≈ 300 ms.

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

Cálculo de Tamaño de Ventana TCP: El servidor anuncia un factor de escala de ventana TCP de 7.

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

El tamaño de la ventana TCP del servidor es de ≈ 1600000 bytes:

No.	Time	Source	Destination	Protocol	Length	Window size value	Calculated window size	Info
23_	0.002579	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [FIN, ACK]
23_	0.266847	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23_	0.268089	172.25.18.134	10.11.2.124	SSHv2	198	12854	1645312	Server: Encrypted pack
23_	0.000076	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pack
23_	0.000351	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pack
23_	0.000092	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23_	0.000015	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23_	0.000091	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=

En función de estos valores, la fórmula de producto de retraso de ancho de banda ofrece:

$$1600000 * 8 / 0,3 = \text{velocidad máxima de transferencia teórica de 43 Mbps}$$

Caso 6. Transferencia TCP lenta (situación 2)

Descripción del problema: La transferencia de archivos FTP (descarga) a través del firewall es lenta.

Esta imagen muestra la topología:



Flujo afectado:

Src IP: 192.168.2.220

Dst IP: 192.168.1.220

Protocolo: FTP

Análisis de captura

Activar capturas en el motor LINA de FTD.

```
<#root>
```

```
firepower#
```

```
capture CAPI type raw-data buffer 33554432 interface INSIDE match tcp host 192.168.2.220 host 192.168.1.220
```

```
firepower#
```

```
cap CAPO type raw-data buffer 33554432 interface OUTSIDE match tcp host 192.168.2.220 host 192.168.1.220
```

Seleccione un paquete FTP-DATA y siga el canal de datos FTP en captura FTD INSIDE (CAPI):

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	0.00061	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	15mb)

El contenido de la secuencia FTP-DATA:

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	1.926564	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=3577288526 TSecr=0 WS=128
29	1.981584	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2669998978 Ack=1884231612 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
30	0.000488	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669998979 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
34	0.000167	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=2669998977 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=2669998927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699992175 SRE=26699993423
39	0.000077	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669998927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699992175 SRE=26699994671
40	0.389096	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669998927 Ack=1884231612 Win=60848 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=26699994671 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
42	0.000489	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
43	0.000045	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
44	0.000077	192.168.1.220	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=26699995919 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
46	0.000030	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=26699995919 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=26699997167 SRE=2669999663
47	0.000504	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=26699995919 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=26699997167 SRE=2670000911
49	0.918126	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=26699995919 Ack=1884231612 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
50	0.000900	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670000911 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
51	0.000519	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
52	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	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)
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	0.000166	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=26700007151 SRE=26700007151
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=2670000655 SRE=26700008399
60	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

El contenido de captura de CAPO:

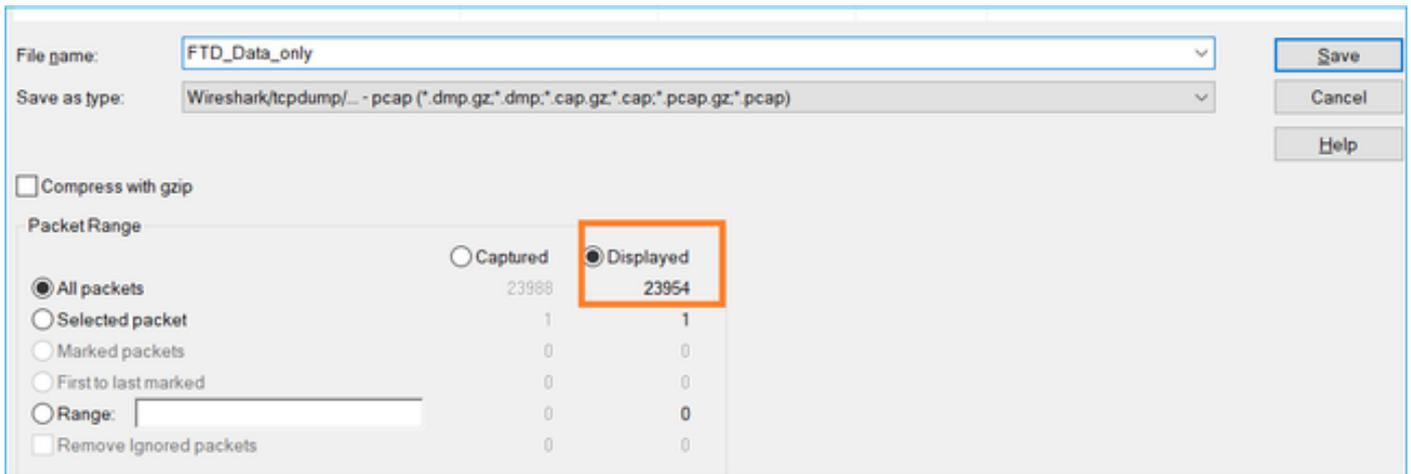
31	0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
33	1.926534	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288526 TSecr=0 WS=128
34	1.981400	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2224318161 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=2224318161 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
38	0.001328	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
40	0.000641	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
41	0.000381	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
42	0.000046	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	0.918010	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)

Puntos clave:

1. Hay paquetes TCP Out-Of-Order (OOO).
2. Hay una retransmisión TCP.
3. Hay una indicación de pérdida de paquetes (paquetes perdidos).



Sugerencia: guarde las capturas mientras navega hasta Archivo > Exportar paquetes especificados. A continuación, guarde sólo el intervalo de paquetes mostrado



Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Identifique la ubicación de pérdida de paquetes.

En casos como este, debe tomar capturas simultáneas y utilizar la metodología de dividir y conquistar para identificar los segmentos de red que causan la pérdida de paquetes. Desde el punto de vista del firewall, existen tres escenarios principales:

1. La pérdida de paquetes es causada por el propio firewall.
2. La pérdida de paquetes se produce después del dispositivo de firewall (dirección del servidor al cliente).
3. La pérdida de paquetes se produce en dirección ascendente al dispositivo de firewall (dirección del cliente al servidor).

Pérdida de paquetes causada por el firewall: para identificar si la pérdida de paquetes es causada por el firewall, es necesario comparar la captura de ingreso con la captura de egreso. Hay muchas maneras de comparar 2 capturas diferentes. En esta sección se muestra una forma de realizar esta tarea.

Procedimiento para comparar 2 capturas con el fin de identificar la pérdida de paquetes

Paso 1. Asegúrese de que las 2 capturas contengan paquetes de la misma ventana de tiempo. Esto significa que no debe haber paquetes en una captura que fueron capturados antes o después de la otra captura. Hay algunas formas de hacerlo:

- Verifique los valores de identificación IP (ID) del primer y último paquete.
- Verifique los valores de la marca de tiempo del primer y último paquete.

En este ejemplo puede ver que los primeros paquetes de cada captura tienen los mismos valores de ID de IP:

No.	Time	Source	Destination	Protocol	Length	Identification	Info
1	2019-10-16 16:13:44.169394	192.168.2.220	192.168.1.220	TCP	74	0x0a34 (2612)	54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=357728500 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=357728526 TSecr=0 WS=128
3	2019-10-16 16:13:47.177542	192.168.2.220	192.168.1.220	TCP	74	0x151f (5407)	2388 → 54494 [SYN, ACK] Seq=2669989678 Ack=1884231612 Win=8192 Len=0 MSS=1260 MS=256 SACK_PERM=1 TSval=4264384 TSecr=357728500
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	192.168.2.220	192.168.1.220	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	192.168.2.220	192.168.1.220	TCP	1314	0x1522 (5410)	
8	2019-10-16 16:13:47.180915	192.168.2.220	192.168.1.220	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	192.168.2.220	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	192.168.2.220	192.168.1.220	TCP	1314	0x1526 (5414)	
14	2019-10-16 16:13:47.490910	192.168.2.220	192.168.2.220	TCP	1314	0x1528 (5416)	
15	2019-10-16 16:13:47.490987	192.168.2.220	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	192.168.2.220	192.168.1.220	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	192.168.2.220	192.168.1.220	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	192.168.2.220	192.168.1.220	TCP	1314	0x152f (5423)	
23	2019-10-16 16:13:48.411630	192.168.2.220	192.168.1.220	TCP	1314	0x1530 (5424)	
24	2019-10-16 16:13:48.411645	192.168.2.220	192.168.2.220	TCP	1314	0x1532 (5426)	
25	2019-10-16 16:13:48.411660	192.168.2.220	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)	

En caso de que no sean iguales, entonces:

1. Compare las marcas de tiempo del primer paquete de cada captura.
2. Desde la captura con la última marca de tiempo, obtenga un filtro de ella, cambie el filtro de marca de tiempo de == a >= (el primer paquete) y <= (el último paquete), p. ej.:

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

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

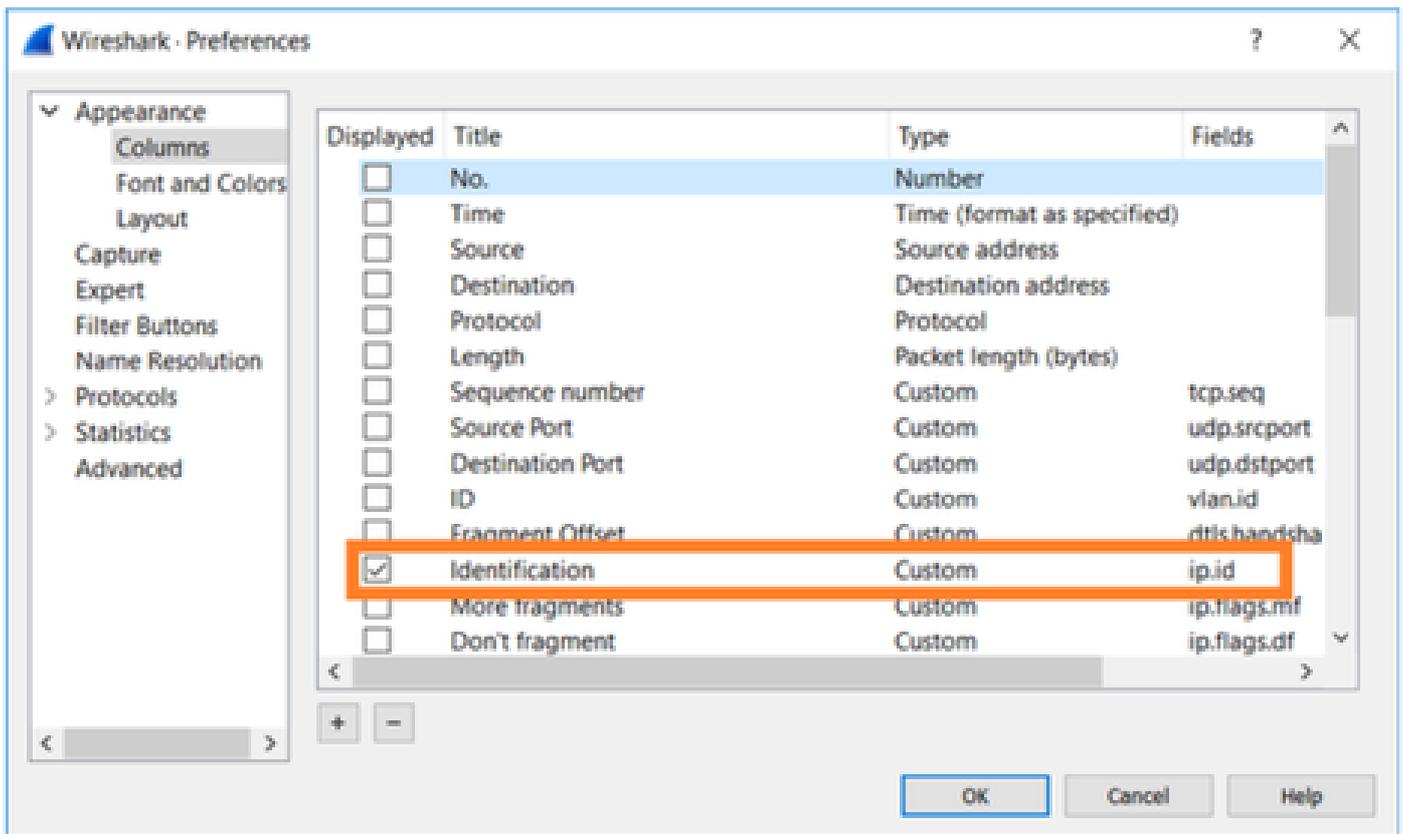
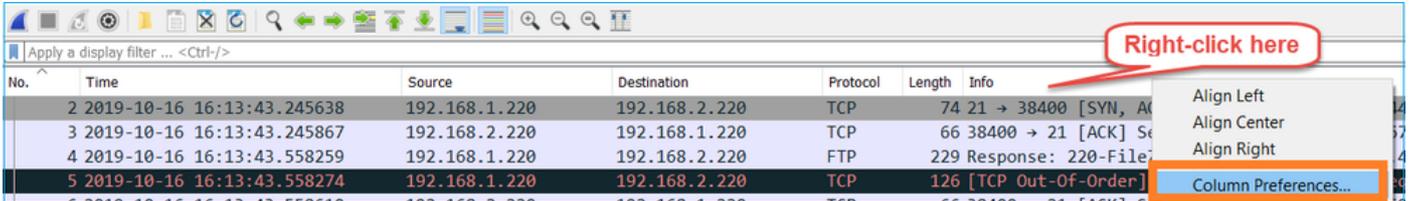
3. Exporte los paquetes especificados a una nueva captura, seleccione Archivo > Exportar paquetes especificados y guarde los paquetes mostrados. En este punto, ambas capturas deben contener paquetes que cubran la misma ventana de tiempo. Ahora puede iniciar la comparación de las 2 capturas.

Paso 2. Especifique qué campo de paquete se utiliza para la comparación entre las 2 capturas. Ejemplo de campos que se pueden utilizar:

- Identificación de IP
- Número de secuencia RTP

- Número de secuencia ICMP

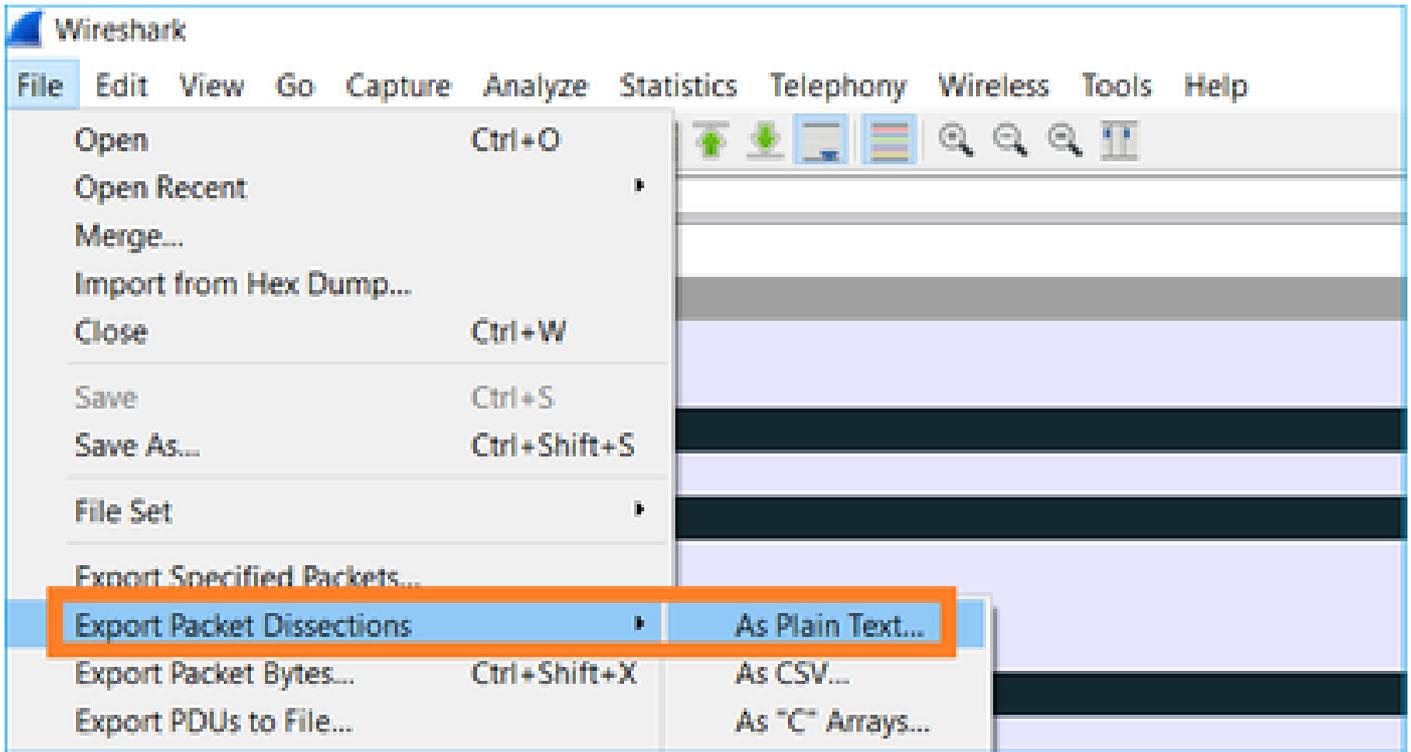
Cree una versión de texto de cada captura que contenga el campo para cada paquete especificado en el paso 1. Para hacer esto, deje solamente la columna de interés, por ejemplo, si desea comparar paquetes basados en la identificación IP, modifique la captura como se muestra en la imagen.



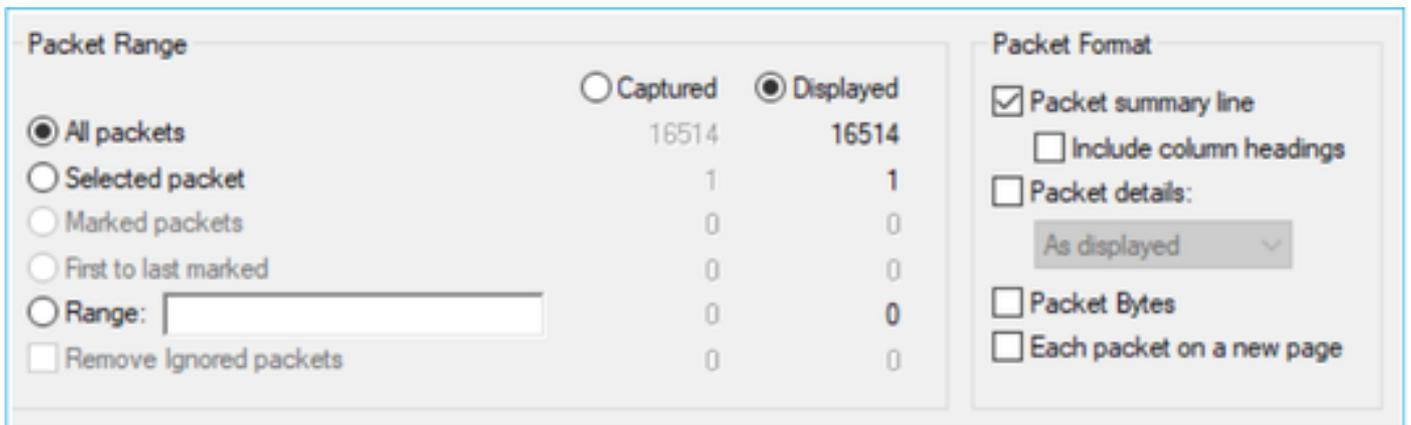
El resultado:

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

Paso 3. Cree una versión de texto de la captura (Archivo > Exportar disecciones de paquetes > Como texto sin formato...), como se muestra en la imagen:



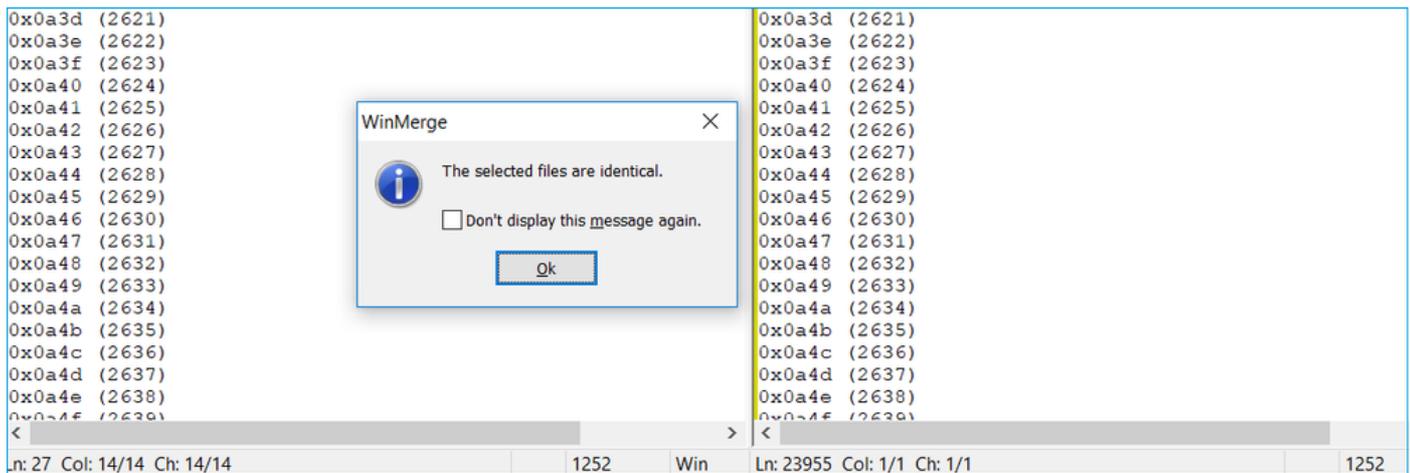
Desactive las opciones Incluir encabezados de columna y Detalles de paquete para exportar sólo los valores del campo mostrado, como se muestra en la imagen:



Paso 4. Ordene los paquetes de los archivos. Puede utilizar el comando sort de Linux para hacer esto:

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

Paso 5. Utilice una herramienta de comparación de texto (por ejemplo, WinMerge) o el comando Linux diff para encontrar las diferencias entre las 2 capturas.



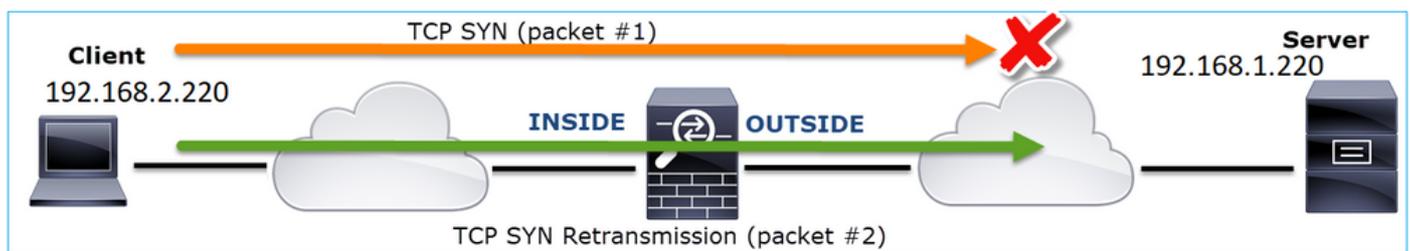
En este caso, la captura de CAPI y CAPO para el tráfico de datos FTP es idéntica. Esto prueba que la pérdida de paquetes no fue causada por el firewall.

Identifique la pérdida de paquetes de flujo ascendente/descendente.

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=3577288500 TSecr=0 WS=1
3	2019-10-16 16:13:47.177450	192.168.1.220	192.168.2.220	TCP	74	2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=3577291508 TSecr=4264384
4	2019-10-16 16:13:47.178060	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
5	2019-10-16 16:13:47.179388	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224316912 Ack=2157030682 Win=66048 Len=1248 TSval=4264384 TSecr=3577291508
6	2019-10-16 16:13:47.180029	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
7	2019-10-16 16:13:47.180410	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224319408 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291510
8	2019-10-16 16:13:47.180456	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224320656 Ack=2157030682 Win=66048 Len=1248 TSval=4264384 TSecr=3577291510
9	2019-10-16 16:13:47.180746	192.168.2.220	192.168.1.220	TCP	78	[TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=35072 Len=0 TSval=3577291510 TSecr=4264415
10	2019-10-16 16:13:47.180822	192.168.2.220	192.168.1.220	TCP	78	[TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=37888 Len=0 TSval=3577291510 TSecr=4264415
11	2019-10-16 16:13:47.489827	192.168.1.220	192.168.2.220	TCP	1314	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224318160 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
12	2019-10-16 16:13:47.490407	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
13	2019-10-16 16:13:47.490819	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224321904 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
14	2019-10-16 16:13:47.490880	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224322400 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
15	2019-10-16 16:13:47.490956	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224325648 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
16	2019-10-16 16:13:47.491246	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415

Puntos clave:

1. Este paquete es una retransmisión TCP. Específicamente, es un paquete TCP SYN enviado desde el cliente al servidor para datos FTP en modo pasivo. Dado que el cliente reenvía el paquete y puede ver el SYN (paquete #1) inicial, el paquete se perdió en dirección ascendente hacia el firewall.



En este caso, existe la posibilidad de que el paquete SYN llegara al servidor, pero el paquete SYN/ACK se perdió en el camino de regreso:



2. Hay un paquete del servidor y Wireshark identificó que el segmento anterior no fue visto/capturado. Dado que el paquete no capturado se envió desde el servidor al cliente y no se vio en la captura del firewall, esto significa que el paquete se perdió entre el servidor y el firewall.



Esto indica que hay pérdida de paquetes entre el servidor FTP y el firewall.

Acción 2. Tome Capturas Adicionales.

Realice capturas adicionales junto con capturas en los terminales. Intente aplicar el método divide y vencerás para aislar aún más el segmento problemático que causa la pérdida de paquetes.

No.	Time	Source	Destination	Protocol	Length	Info
155	2019-10-16 16:13:51.749845	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
156	2019-10-16 16:13:51.749860	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
157	2019-10-16 16:13:51.749872	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
158	2019-10-16 16:13:51.750722	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224385552 Win=180480 Len=0 TSv
159	2019-10-16 16:13:51.750744	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
160	2019-10-16 16:13:51.750768	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800 Win=183424 Len=0 TSv
161	2019-10-16 16:13:51.750782	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
162	2019-10-16 16:13:51.751001	192.168.2.220	192.168.1.220	TCP	70	[TCP Dup ACK 160#1] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
163	2019-10-16 16:13:51.751024	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
164	2019-10-16 16:13:51.751378	192.168.2.220	192.168.1.220	TCP	70	[TCP Dup ACK 160#2] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
165	2019-10-16 16:13:51.751402	192.168.1.220	192.168.2.220	FTP-DA..	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
166	2019-10-16 16:13:51.751622	192.168.2.220	192.168.1.220	TCP	70	[TCP Dup ACK 160#3] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
167	2019-10-16 16:13:51.751648	192.168.1.220	192.168.2.220	FTP-DA..	1314	[TCP Fast Retransmission] FTP Data: 1248 bytes (PASV) (RETR file15mb)

< Frame 167: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits) on interface 0
 > Ethernet II, Src: Vmware_30:2b:78 (00:0c:29:30:2b:78), Dst: Cisco_9d:89:9b (50:3d:e5:9d:89:9b)
 > Internet Protocol Version 4, Src: 192.168.1.220, Dst: 192.168.2.220
 > Transmission Control Protocol, Src Port: 2388, Dst Port: 54494, 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)

Puntos clave:

1. El receptor (el cliente FTP en este caso) rastrea los números de secuencia TCP entrantes. Si detecta que se ha omitido un paquete (se omitió un número de secuencia esperado), genera un paquete ACK con el ACK='número de secuencia esperado que se omitió'. En este ejemplo, Ack=2224386800.

2. El ACK de duplicación activa una retransmisión rápida TCP (retransmisión dentro de los 20 mseg después de que se reciba un ACK duplicado).

¿Qué significan las ACK duplicadas?

- Algunos ACK duplicados, pero ninguna retransmisión real, indican que es más probable que haya paquetes que lleguen fuera de servicio.
- Las ACK duplicadas seguidas de retransmisiones reales indican que hay cierta cantidad de pérdida de paquetes.

Acción 3. Calcule el tiempo de procesamiento del firewall para los paquetes de tránsito.

Aplique la misma captura en 2 interfaces diferentes:

```
<#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
```

Exporte la verificación de captura para comprobar la diferencia de tiempo entre los paquetes de ingreso y egreso

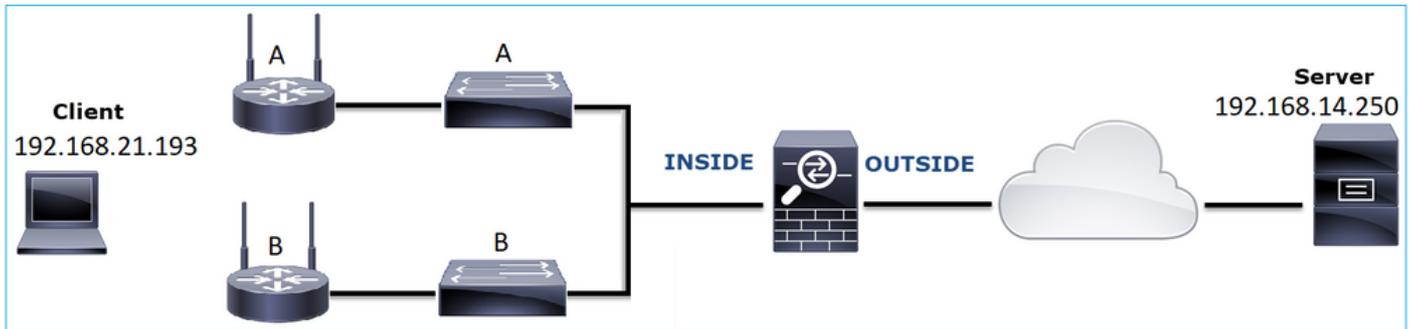
Caso 7. Problema de conectividad TCP (corrupción de paquetes)

Descripción de problemas:

El cliente inalámbrico (192.168.21.193) intenta conectarse a un servidor de destino (192.168.14.250 - HTTP) y existen dos situaciones diferentes:

- Cuando el cliente se conecta al punto de acceso (AP) 'A', la conexión HTTP no funciona.
- Cuando el cliente se conecta al punto de acceso (AP) 'B', la conexión HTTP funciona.

Esta imagen muestra la topología:



Flujo afectado:

Src IP: 192.168.21.193

Dst IP: 192.168.14.250

Protocolo: TCP 80

Análisis de captura

Habilitar capturas en el motor LINA de FTD:

```
<#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
```

Capturas - Escenario funcional:

Como base, siempre es muy útil tener capturas de un escenario de funcionalidad comprobada.

Esta imagen muestra la captura realizada en la interfaz NGFW INSIDE

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

Esta imagen muestra la captura realizada en la interfaz EXTERNA de NGFW.

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

Puntos clave:

1. Las 2 capturas son casi idénticas (considere la aleatorización ISN).
2. No hay indicios de pérdida de paquetes.
3. Sin paquetes fuera de servicio (OOO)
4. Hay 3 solicitudes GET HTTP. El primero recibe un mensaje de redirección 404 'No encontrado', el segundo obtiene un 200 'OK' y el tercero recibe un mensaje de redirección 304 'No modificado'.

Capturas: situación de fallo conocido:

El contenido de captura de ingreso (CAPI).

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

Puntos clave:

1. Existe un protocolo de enlace TCP de 3 vías.
2. Hay retransmisiones TCP e indicaciones de pérdida de paquetes.
3. Wireshark identifica un paquete (TCP ACK) como Malformado.

Esta imagen muestra el contenido de captura de salida (CAPO).

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

Puntos clave:

Las 2 capturas son casi idénticas (considere la aleatorización ISN):

1. Existe un protocolo de enlace TCP de 3 vías.
2. Hay retransmisiones TCP e indicaciones de pérdida de paquetes.
3. Wireshark identifica un paquete (TCP ACK) como Malformado.

Verifique el paquete mal formado:

The screenshot shows a Wireshark capture of three network packets. The first two are standard TCP SYN and SYN-ACK packets. The third packet, at time 2013-08-08 15:33:31.913267, is a TCP ACK packet with a length of 2 bytes, which is marked as 'Malformed Packet' (1). The packet details pane shows the following information:

- Source Port: 3072
- Destination Port: 80
- Sequence number: 4231766829
- Acknowledgment number: 867575960
- Flags: 0x010 (ACK)
- Window size value: 65535
- Checksum: 0x01bf [unverified]
- Urgent pointer: 0
- Timestamps
- TCP payload (2 bytes)

The packet is identified as a 'Malformed Packet: Tunnel Socket' (1). The expert info pane shows: 'Malformed Packet (Exception occurred)' with a severity level of 'Error' and a group of 'Malformed'.

The packet bytes pane shows the following hexadecimal data (4):

```
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 a7 7d 33 b6 .....P:;--3-
0030 28 98 50 10 ff ff 01 bf 00 00 00 00 (-P.....-..)
```

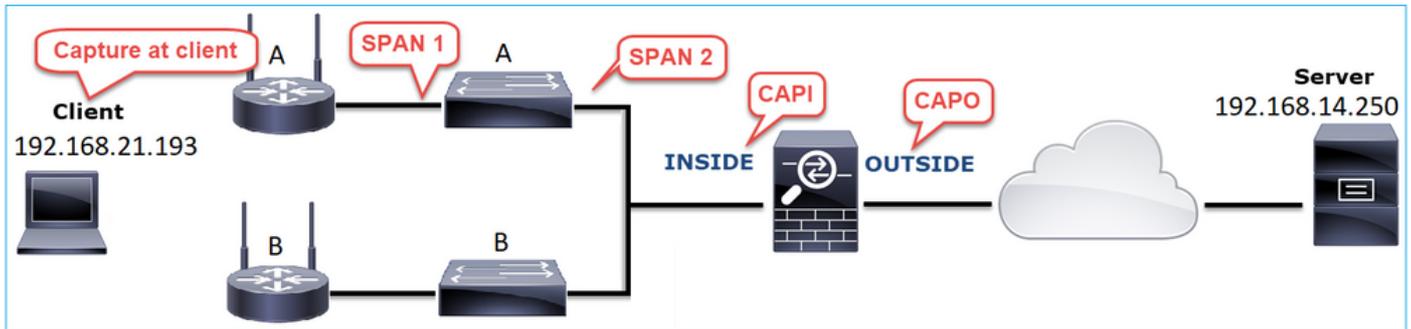
Puntos clave:

1. Wireshark identifica el paquete como Malformado.
2. Tiene una longitud de 2 bytes.
3. Hay una carga útil TCP de 2 bytes.
4. La carga útil es de 4 ceros adicionales (00 00).

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Realice capturas adicionales. Incluya capturas en los puntos finales y, si es posible, intente aplicar el método de división y conquista para aislar el origen de la corrupción del paquete, por ejemplo:

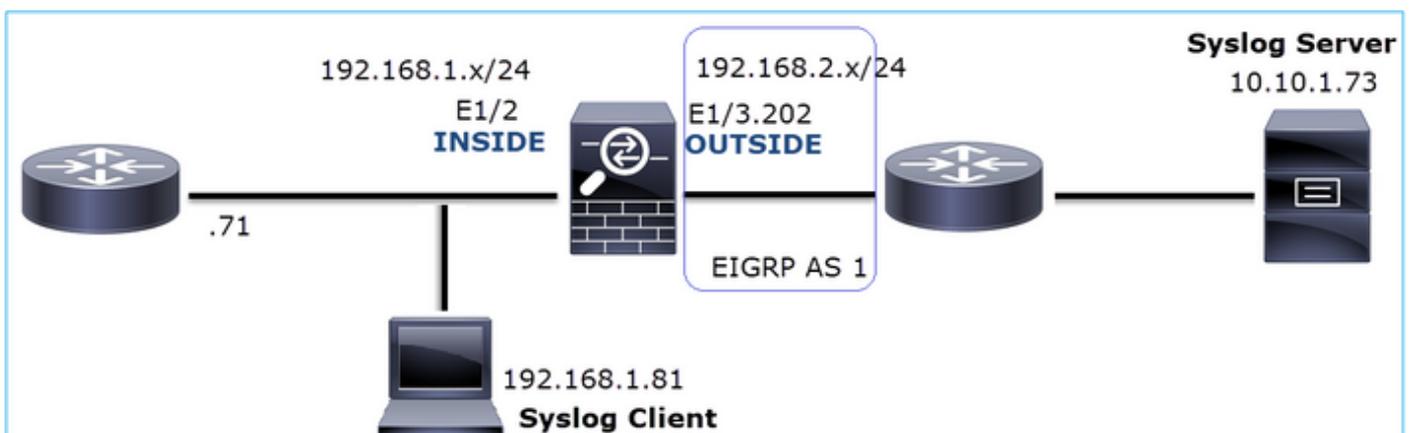


En este caso, los 2 bytes adicionales fueron agregados por el controlador de interfaz 'A' del switch y la solución fue reemplazar el switch que causa la corrupción.

Caso 8. Problema de conectividad UDP (paquetes faltantes)

Descripción del problema: los mensajes de Syslog (UDP 514) no se ven en el servidor de Syslog de destino.

Esta imagen muestra la topología:



Flujo afectado:

Src IP: 192.168.1.81

Dst IP: 10.10.1.73

Protocolo: UDP 514

Análisis de captura

Habilitar capturas en el motor LINA de FTD:

```
<#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
```

Las capturas de FTD no muestran paquetes:

```
<#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 trace interface OUTSIDE [Capturing - 0 bytes]
  match udp host 192.168.1.81 host 10.10.1.73 eq syslog
```

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Compruebe la tabla de conexiones FTD.

Para comprobar una conexión específica, puede utilizar esta sintaxis:

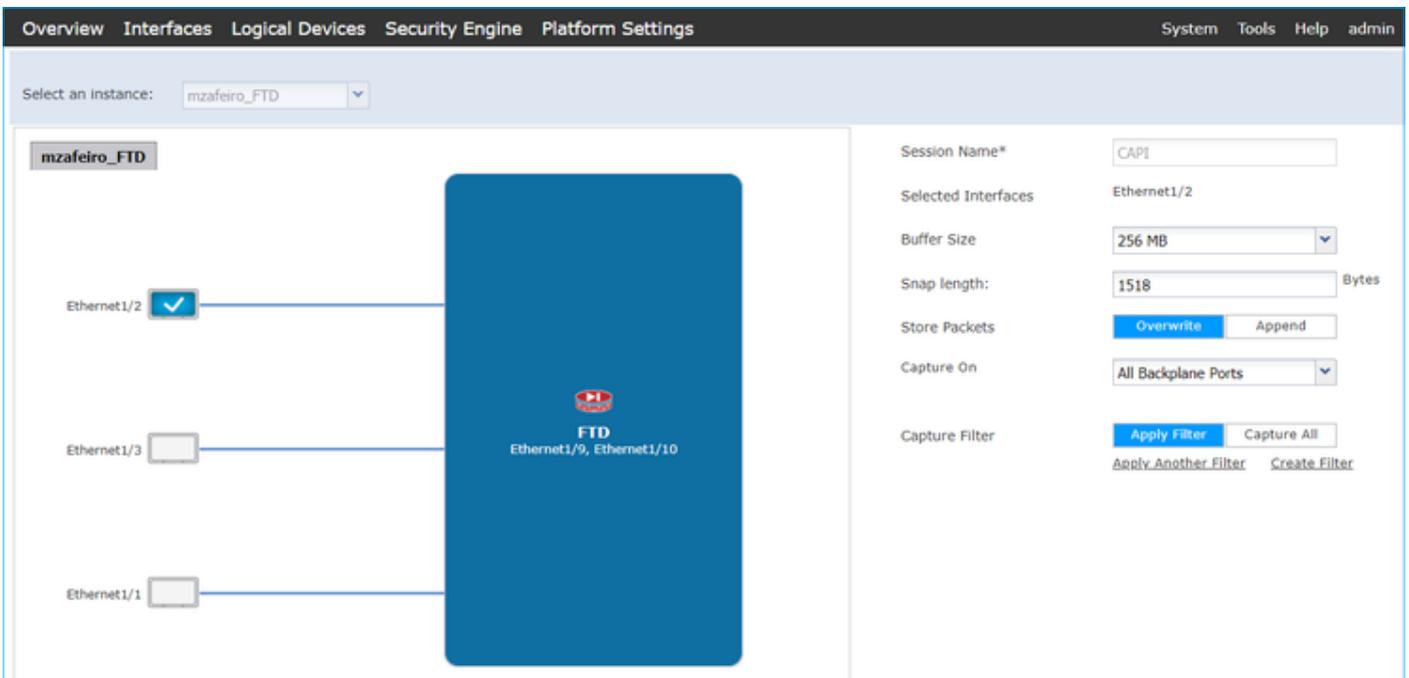
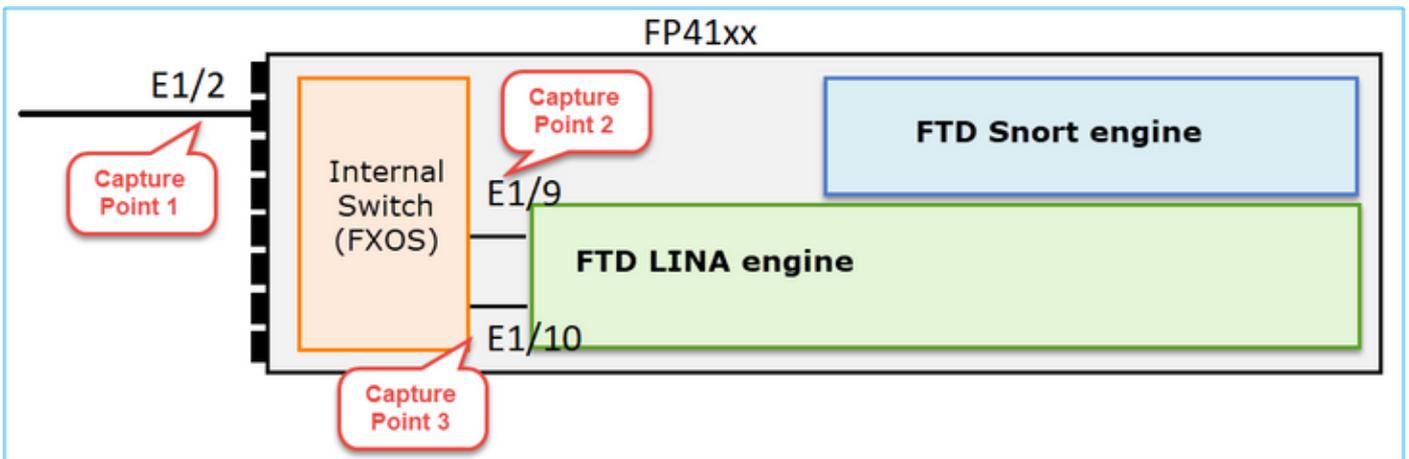
```
<#root>
firepower#
show conn address 192.168.1.81 port 514
10 in use, 3627189 most used
Inspect Snort:
  preserve-connection: 6 enabled, 0 in effect, 74 most enabled, 0 most in effect
UDP
INSIDE
  10.10.1.73:514
INSIDE
  192.168.1.81:514, idle 0:00:00, bytes
480379697
, flags -
o
N1
```

Puntos clave:

1. Las interfaces de entrada y salida son las mismas (giro en U).
2. El número de bytes tiene un valor significativamente grande (~5 GBytes).
3. El indicador "o" indica descarga de flujo (flujo acelerado de hardware). Esta es la razón por la que las capturas de FTD no muestran ningún paquete. La descarga de flujo solo se admite en las plataformas 41xx y 93xx. En este caso, el dispositivo es un 41xx.

Acción 2. Realice capturas a nivel de chasis.

Conéctese al administrador del chasis Firepower y habilite la captura en la interfaz de ingreso (E1/2 en este caso) y en las interfaces de la placa posterior (E1/9 y E1/10), como se muestra en la imagen:



Después de unos segundos:

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

 Sugerencia: en Wireshark, excluya los paquetes etiquetados con VN para eliminar la duplicación de paquetes en el nivel de interfaz física

Antes:

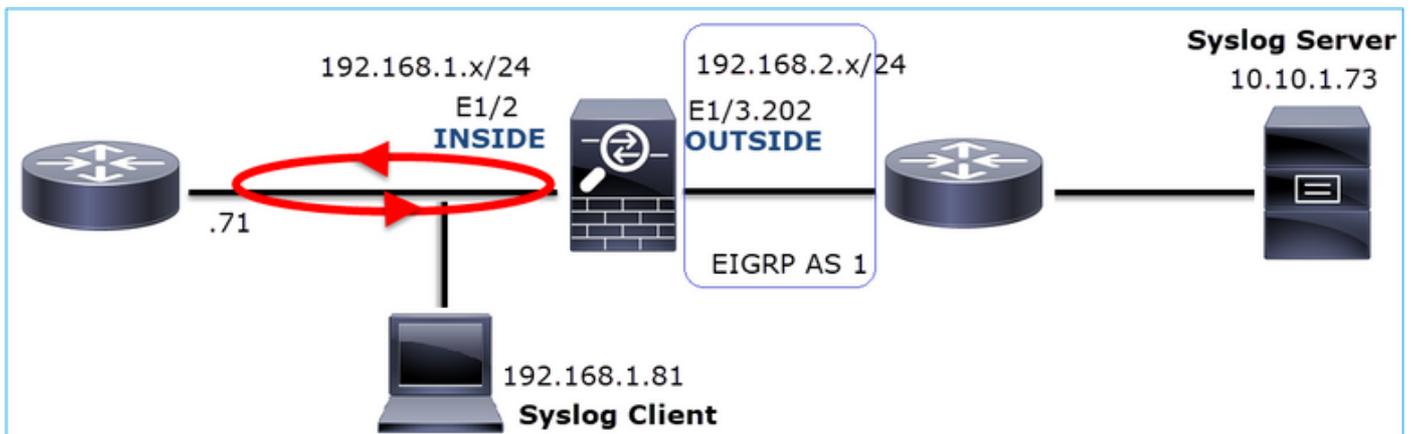
No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
2	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
3	0.0532	Vmware_85:4f:ca	Broadcast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

Después de:

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 identity:192.168.1.81 dur
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

Puntos clave:

1. Se aplica un filtro de visualización para quitar duplicados de paquetes y mostrar sólo los registros del sistema.
2. La diferencia entre los paquetes está en el nivel de microsegundos. Esto indica una velocidad de paquetes muy alta.
3. El valor de Tiempo de vida (TTL) disminuye de forma continua. Esto indica un loop de paquete.



Acción 3. Utilice packet-tracer.

Dado que los paquetes no atraviesan el motor LINA del firewall, no puede realizar un seguimiento activo (captura con seguimiento), pero puede rastrear un paquete emulado con packet-tracer:

```
</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
```

Acción 4. Confirme el enrutamiento de FTD.

Verifique la tabla de ruteo del firewall para ver si hay algún problema de ruteo:

```
<#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
```

Puntos clave:

1. La ruta apunta hacia la interfaz de salida correcta.
2. La ruta se aprendió hace unos minutos (0:02:37).

Acción 5. Confirme el tiempo de actividad de conexión.

Verifique el tiempo de actividad de la conexión para ver cuándo se estableció esta conexión:

```
<#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
```

Punto clave:

1. La conexión se estableció hace aproximadamente 4 minutos (esto es antes de la instalación de la ruta EIGRP en la tabla de ruteo)

Acción 6. Borre la conexión establecida.

En este caso, los paquetes coinciden con una conexión establecida y se rutean a una interfaz de salida incorrecta; esto provoca un loop. Esto se debe al orden de operaciones del firewall:

1. Búsqueda de conexión establecida (esto tiene prioridad sobre la búsqueda de tabla de ruteo global).
2. Búsqueda de traducción de direcciones de red (NAT): la fase UN-NAT (NAT de destino) tiene prioridad sobre PBR y la búsqueda de rutas.
3. Routing basado en políticas (PBR)
4. Búsqueda de tabla de routing global

Dado que la conexión nunca se agota (el cliente Syslog envía paquetes continuamente mientras el tiempo de espera inactivo de la conexión UDP es de 2 minutos), es necesario borrar manualmente la conexión:

```
<#root>
```

```
firepower#
```

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

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

Verifique que se haya establecido una nueva conexión:

```
<#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
```

Acción 7. Configure el tiempo de espera de conexión flotante.

Esta es la solución adecuada para abordar el problema y evitar un ruteo subóptimo, especialmente para los flujos UDP. Vaya a Devices > Platform Settings > Timeout y establezca el valor:

SMTP Server	H.323	Default	0:05:00	(0:0:0 or 0:0:0 - 1193:0:0)
SNMP	SIP	Default	0:30:00	(0:0:0 or 0:5:0 - 1193:0:0)
SSL	SIP Media	Default	0:02:00	(0:0:0 or 0:1:0 - 1193:0:0)
Syslog	SIP Disconnect:	Default	0:02:00	(0:02:0 or 0:0:1 - 0:10:0)
Timeouts	SIP Invite	Default	0:03:00	(0:1:0 or 0:1:0 - 0:30:0)
Time Synchronization	SIP Provisional Media	Default	0:02:00	(0:2:0 or 0:1:0 - 0:30:0)
UCAPL/CC Compliance	Floating Connection	Custom	0:00:30	(0:0:0 or 0:0:30 - 1193:0:0)
	Xlate-PAT	Default	0:00:30	(0:0:30 or 0:0:30 - 0:5:0)

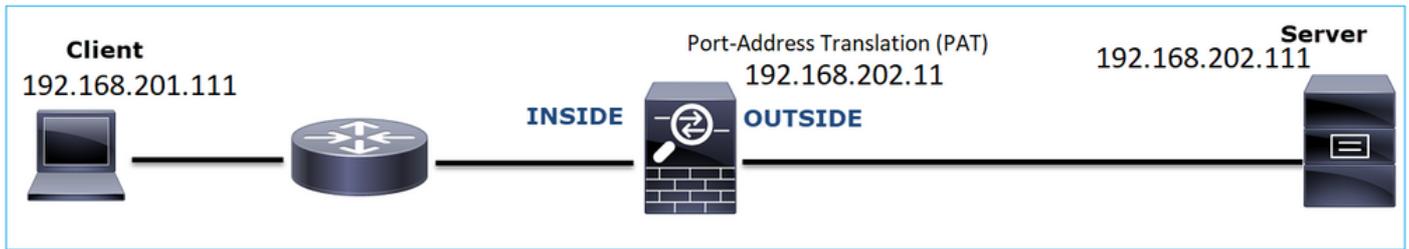
Puede encontrar más detalles sobre el tiempo de espera de conexión flotante en la Referencia de Comandos:

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

Caso 9. Problema de conectividad HTTPS (situación 1)

Descripción del problema: No se puede establecer la comunicación HTTPS entre el cliente 192.168.201.105 y el servidor 192.168.202.101

Esta imagen muestra la topología:



Flujo afectado:

Src IP: 192.168.201.111

Dst IP: 192.168.202.111

Protocolo: TCP 443 (HTTPS)

Análisis de captura

Habilitar capturas en el motor LINA de FTD:

La IP utilizada en la captura OUTSIDE es diferente debido a la configuración de Traducción de Dirección de Puerto .

```
<#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
```

Esta imagen muestra la captura realizada en la interfaz NGFW INSIDE:

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	TLv1	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

Puntos clave:

1. Existe un protocolo de enlace TCP de 3 vías.
2. Se inicia la negociación SSL. El cliente envía un mensaje de saludo de cliente.
3. Se ha enviado un ACK TCP al cliente.
4. Hay un TCP RST enviado al cliente.

Esta imagen muestra la captura realizada en la interfaz EXTERNA de NGFW.

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 (12881)	15880 → 443 [SYN] Seq=2486930707 Min=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 Min=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=15880
35	2018-02-01 10:39:35.189214	192.168.202.11	192.168.202.111	TCP	70	0x2f32 (12882)	15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Min=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 Min=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 Min=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 Min=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	0xdd21 (54817)	15880 → 443 [RST] Seq=2486930895 Min=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 Min=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=15880

Puntos clave:

1. Existe un protocolo de enlace TCP de 3 vías.
2. Se inicia la negociación SSL. El cliente envía un mensaje de saludo de cliente.
3. Hay retransmisiones TCP enviadas desde el firewall hacia el servidor.
4. Hay un TCP RST enviado al servidor.

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Realice capturas adicionales.

Una captura tomada en el servidor revela que el servidor recibió los saludos del cliente TLS con una suma de comprobación TCP dañada y los descarta silenciosamente (no hay RST TCP ni ningún otro paquete de respuesta hacia el cliente):

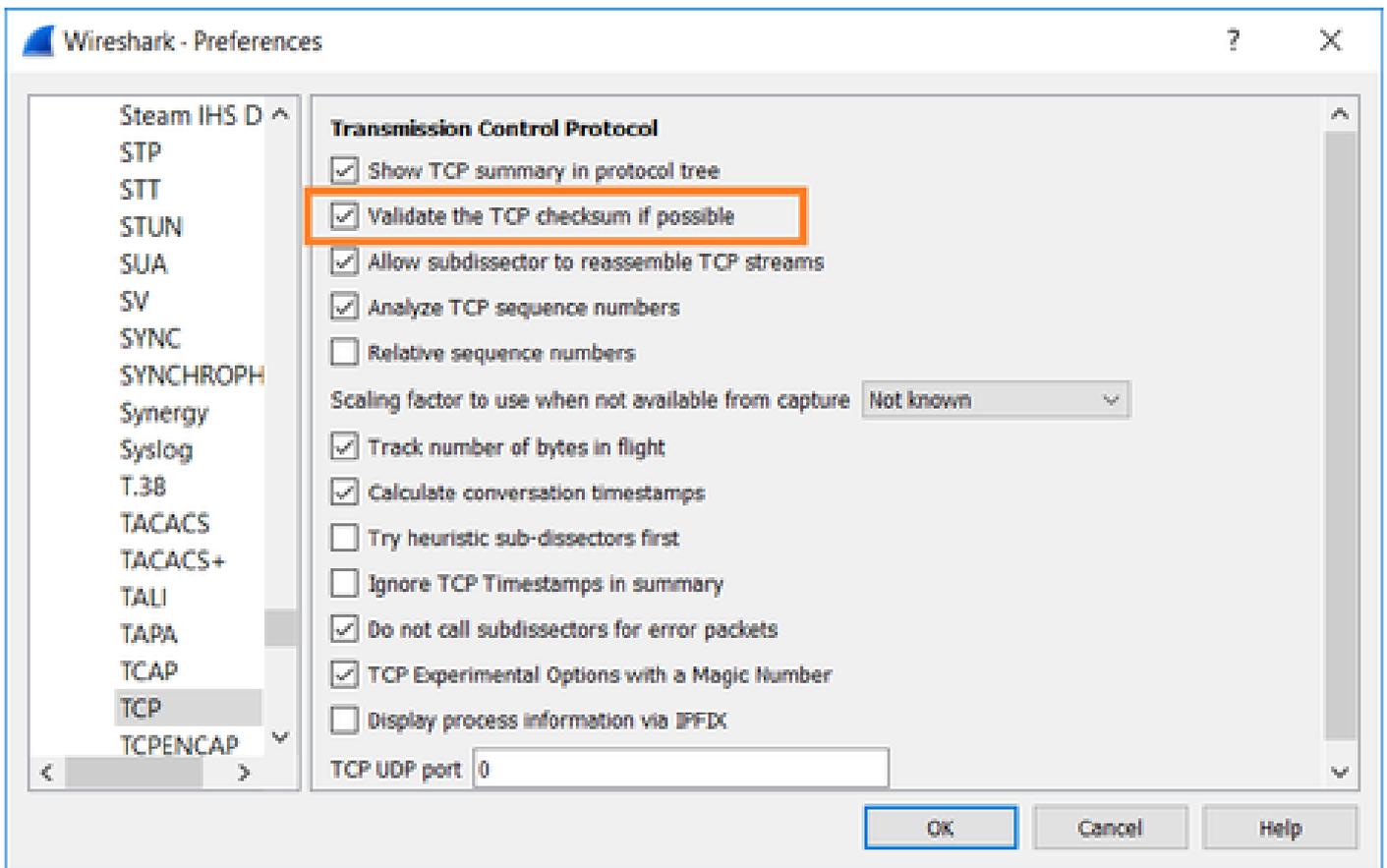
```

21:26:27.133677 IP (tos 0x0, ttl 64, id 52534, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x0c65 (incorrect -> 0x3063), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192658174 ecr 3119615816], length 187
21:26:29.155652 IP (tos 0x0, ttl 64, id 47365, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x4db7 (incorrect -> 0x71b5), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192660198 ecr 0], length 187
21:26:33.178142 IP (tos 0x0, ttl 64, id 34991, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x3dd (incorrect -> 0x61fb), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192664224 ecr 0], length 187
21:26:41.189640 IP (tos 0x0, ttl 64, id 63114, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x1e9 (incorrect -> 0x42a7), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192672244 ecr 0], length 187
21:26:57.195947 IP (tos 0x0, ttl 64, id 54817, offset 0, flags [DF], proto TCP (6), length 52)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [R], cksum 0x9ee (incorrect -> 0xc2e8), seq 2486930895, win 64, options [nop,nop,TS v
al 192688266 ecr 0], length 0
21:26:58.668973 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
  192.168.202.111.443 > 192.168.202.11.15880: Flags [S.], cksum 0x15fb (incorrect -> 0xffd2), seq 3674405382, ack 2486930708, win 28960, o
ptions [mss 1460,sackOK,TS val 3119647415 ecr 192658158,nop,wscale 7], length 0
^C
154 packets captured
154 packets received by filter

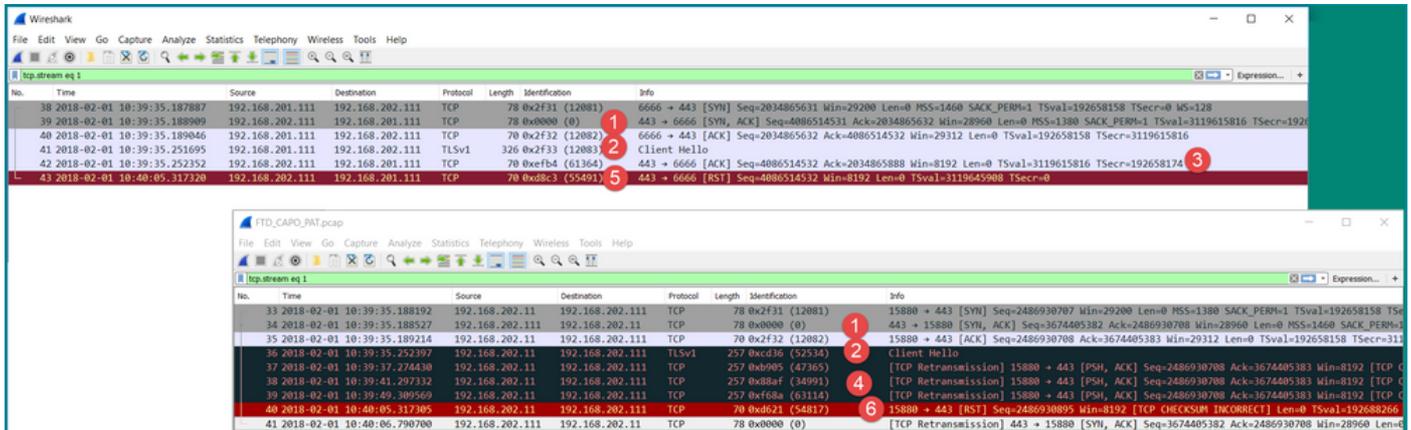
```

Cuando se combinan todos los elementos:

En este caso, para entender, hay una necesidad de habilitar en Wireshark la opción Validar la suma de comprobación TCP si es posible. Vaya a Edit > Preferences > Protocols > TCP, como se muestra en la imagen.



En este caso, es útil poner las capturas lado a lado para obtener la imagen completa:



Puntos clave:

1. Existe un protocolo de enlace TCP de 3 vías. Los ID de IP son los mismos. Esto significa que el flujo no fue procesado como proxy por el firewall.
2. Un saludo de cliente TLS proviene del cliente con ID IP 12083. El paquete es proxy por el firewall (el firewall, en este caso, se configuró con la política de descifrado de TLS) y el ID de IP se cambia a 52534. Además, la suma de comprobación TCP del paquete se daña (debido a un defecto de software que luego se corrige).
3. El firewall se encuentra en el modo de proxy TCP y envía un ACK al cliente (que suplanta al servidor).

```

33 2018-02-01 10:39:35.188192 192.168.202.11 192.168.202.111 TCP 78 0x2f31 (12081) 15880 → 443 [SYN] Seq=2486930707 Min=29200 Len=0 MSS=1380 S
34 2018-02-01 10:39:35.188527 192.168.202.111 192.168.202.11 TCP 78 0x0000 (0) 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Min=29
35 2018-02-01 10:39:35.189214 192.168.202.11 192.168.202.111 TCP 70 0x2f32 (12082) 15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Min=29312 L
36 2018-02-01 10:39:35.252397 192.168.202.11 192.168.202.111 TLSv1 257 0xcd36 (52534) Client Hello

```

```

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

```

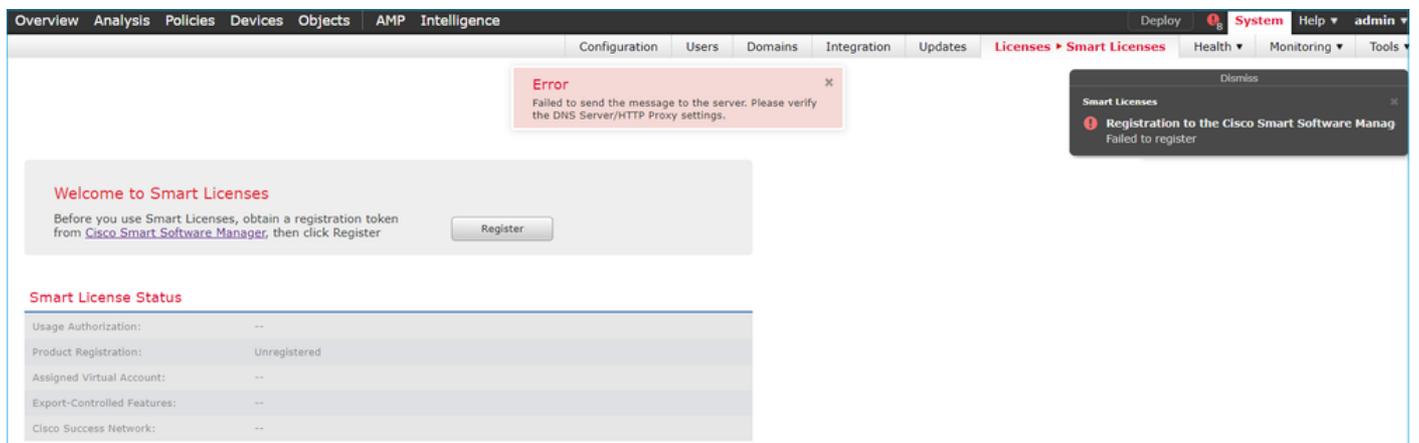
4. El firewall no recibe ningún paquete TCP ACK del servidor y retransmite el mensaje de saludo del cliente TLS. Esto se debe de nuevo al modo de proxy TCP que activó el firewall.
5. Después de ~30 segundos, el firewall se da por vencido y envía un TCP RST hacia el cliente.
6. El firewall envía un TCP RST hacia el servidor.

Para referencia:

[Procesamiento de intercambio de señales Firepower TLS/SSL](#)

Caso 10. Problema de conectividad HTTPS (situación 2)

Descripción del problema: el registro de FMC Smart License falla.



Esta imagen muestra la topología:



Flujo afectado:

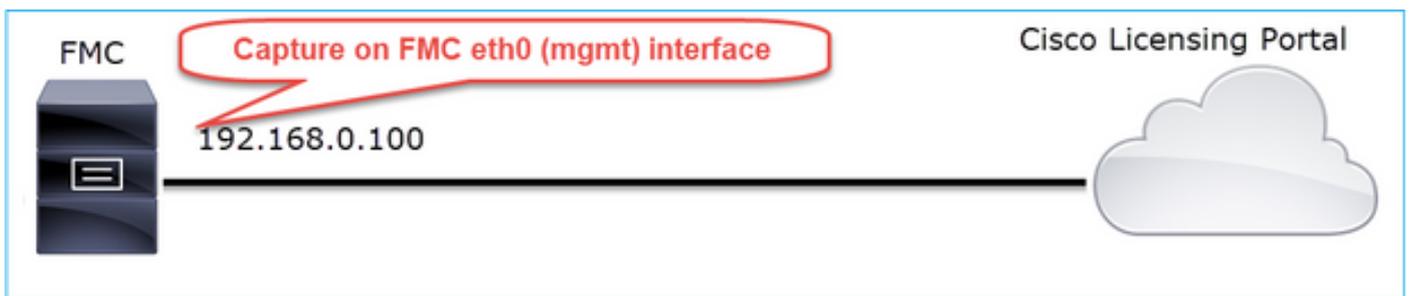
Src IP: 192.168.0.100

Dst: tools.cisco.com

Protocolo: TCP 443 (HTTPS)

Análisis de captura

Habilite la captura en la interfaz de gestión de FMC:



Intente registrarse de nuevo. Cuando aparezca el mensaje de error, presione CTRL-C para detener la captura:

```
<#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
```

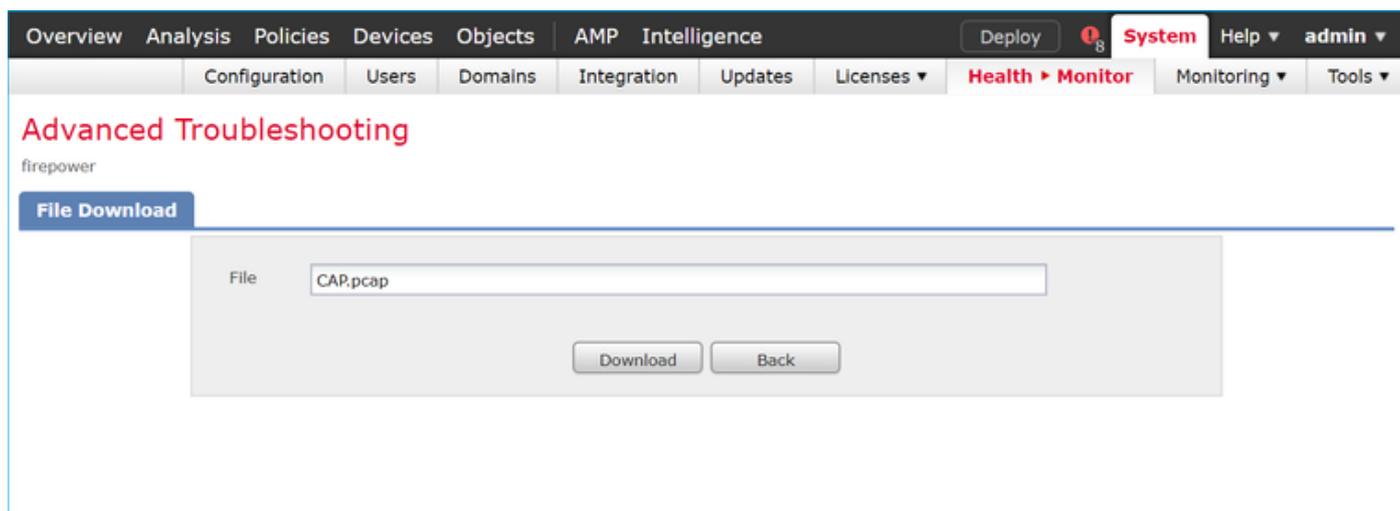
```
<- CTRL-C
```

```
264 packets received by filter
```

```
0 packets dropped by kernel
```

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

Recopile la captura del FMC (System > Health > Monitor, seleccione el dispositivo y seleccione Advanced Troubleshooting), como se muestra en la imagen:



La imagen muestra la captura de FMC en Wireshark:

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 SACK_PERM=1
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=0 MSS=1460 SACK_PERM=1 TSval=2040670996 TSecr=0 WS=128
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

 Sugerencia: Para comprobar todas las sesiones TCP nuevas capturadas, utilice el filtro de visualización `tcp.flags==0x2` en Wireshark. Esto filtra todos los paquetes SYN TCP que fueron capturados.

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

 Sugerencia: Aplique como columna el campo Nombre del servidor del saludo de SSL Client.

75 2019-10-23 07:45:14.634091 192.168.0.100 72.163.4.38 TLSv1.2 571 Client Hello

> Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits)
 > Ethernet II, Src: Vmware_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38
 > Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 517

Secure Sockets Layer

- TLsv1.2 Record Layer: Handshake Protocol
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 512
 - Handshake Protocol: Client Hello
 - Handshake Type: Client Hello (1)
 - Length: 508
 - Version: TLS 1.2 (0x0303)
 - Random: 234490a107438c73b595646532
 - Session ID Length: 0
 - Cipher Suites Length: 100
 - Cipher Suites (50 suites)
 - Compression Methods Length: 1
 - Compression Methods (1 method)
 - Extensions Length: 367
 - Extension: server_name (len=20)
 - Type: server_name (0)
 - Length: 20
 - Server Name Indication extension
 - Server Name list length: 18
 - Server Name Type: host_name (0)
 - Server Name length: 15
 - Server Name: tools.cisco.com

Context menu options: Expand Subtrees, Collapse Subtrees, Expand All, Collapse All, **Apply as Column**, Apply as Filter, Prepare a Filter, Conversation Filter, Colorize with Filter, Follow, Copy, Show Packet Bytes..., Export Packet Bytes..., Wiki Protocol Page, Filter Field Reference, Protocol Preferences, Decode As..., Go to Linked Packet, Show Linked Packet in New Window

Sugerencia: aplique este filtro de visualización para ver sólo los mensajes de saludo del cliente `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

Nota: En el momento de escribir este documento, el portal de licencias inteligentes (tools.cisco.com) utiliza las siguientes direcciones IP: 72.163.4.38, 173.37.145.8

Siga uno de los flujos TCP (Follow > TCP Stream), como se muestra en la imagen.

75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.co
111	2019-10-23 07:45:25.136089	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.co
140	2019-10-23 07:45:35.637252	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.co
166	2019-10-23 07:45:46.136858	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.co
195	2019-10-23 07:45:56.635438	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.co
230	2019-10-23 07:46:07.221567	10.229.20.96	192.168.0.100	TLSv1.2	571	
240	2019-10-23 07:46:07.228486	10.229.20.96	192.168.0.100	TLSv1.2	571	

Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface eth0, 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: 571 Secure Sockets Layer

TLsv1.2 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 512

- Mark/Unmark Packet
- Ignore/Unignore Packet
- Set/Unset Time Reference
- Time Shift...
- Packet Comment...
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversion Filter
- Colorize Conversion
- SCTP
- Follow
 - TCP Stream
 - UDP Stream
 - SSL Stream
 - HTTP Stream
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74		35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16127801 TSecr=0 WS=128
73	2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [SYN, ACK] Seq=2770078884 Ack=2427943532 Win=8190 Len=0 MSS=1330
74	2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427943532 Ack=2770078885 Win=29200 Len=0
75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
76	2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=32768 Len=0
77	2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150		Server Hello
78	2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=29200 Len=0
79	2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1304		443 → 35752 [PSH, ACK] Seq=2770078981 Ack=2427944049 Win=32768 Len=1330 [TCP segment of a reassembled PDU]
80	2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=31920 Len=0
81	2019-10-23 07:45:14.966877	72.163.4.38	192.168.0.100	TLSv1.2	155		Certificate
82	2019-10-23 07:45:14.966887	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 07:45:14.967201	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [RST, ACK] Seq=2427944056 Ack=2770080421 Win=0 Len=0
87	2019-10-23 07:45:14.967282	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770080421 Ack=2427944056 Win=32768 Len=0
88	2019-10-23 07:45:14.967398	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [RST] Seq=2427944056 Win=0 Len=0

> Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface eth0
> Ethernet II, Src: Vmware_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
> Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38
> Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 517
Secure Sockets Layer

TLsv1.2 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 512

Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 508
Version: TLS 1.2 (0x0303)
Random: 234490a107438c73b58564653271c7c09fbb7ac16897184...
Session ID Length: 0
Cipher Suites Length: 100
Cipher Suites (50 suites)

Puntos clave:

1. Existe un protocolo de enlace TCP de 3 vías.
2. El cliente (FMC) envía un mensaje de saludo del cliente SSL al portal de Smart Licensing.
3. El ID de sesión SSL es 0. Esto significa que no se ha reanudado el período de sesiones.
4. El servidor de destino responde con los mensajes Hello de servidor, Certificate y Hello Done de servidor.
5. El cliente envía una alerta de error SSL que se refiere a una "CA desconocida".
6. El cliente envía un TCP RST para cerrar la sesión.
7. La duración total de la sesión TCP (desde el establecimiento hasta el cierre) fue de ~0,5 s.

Seleccione el Certificado de servidor y expanda el campo emisor para ver commonName. En este caso, el nombre común revela un dispositivo que ejecuta la función Man-in-the-middle (MITM).

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-sto
                signedCertificate
                  version: v3 (2)
                  serialNumber: 0x00aa23af5d607e00002f423880
                  > signature (sha256WithRSAEncryption)
                    > issuer: rdnSequence (0)
                      > rdnSequence: 3 items (id-at-commonName=FTD4100_MITM,id-at-organizationalUnitName=FTD_OU,id-at-organizationName=FTD_O)
                        > RDNSSequence item: 1 item (id-at-organizationName=FTD_O)
                        > RDNSSequence item: 1 item (id-at-organizationalUnitName=FTD_OU)
                        > RDNSSequence item: 1 item (id-at-commonName=FTD4100_MITM)
                  > validity
                  > subject: rdnSequence (0)
                  > subjectPublicKeyInfo
                > extensions: 6 items
  
```

Esto se muestra en esta imagen:

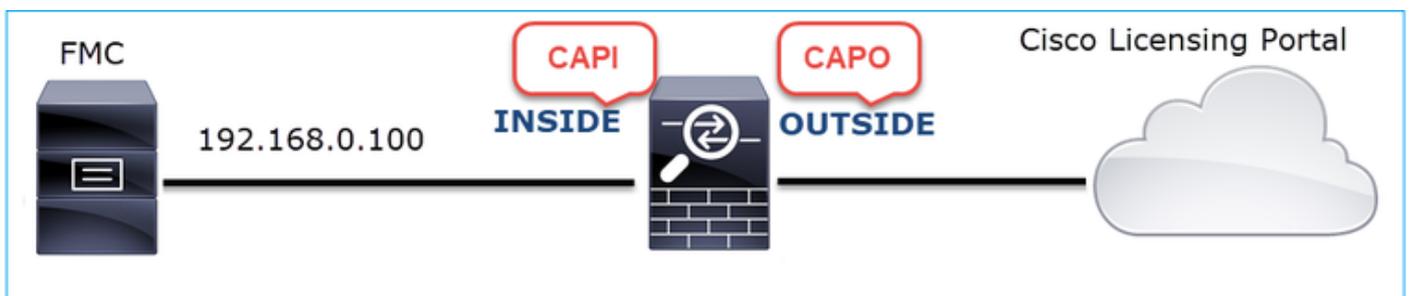


Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Realice capturas adicionales.

Realizar capturas en el dispositivo de firewall de tránsito:



CAPI muestra:

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1221	2019-10-22 17:49:03.212681	192.168.0.100	173.37.145.8	TCP	74		39924 → 443 [SYN] Seq=427175838 Win=29200 Len=0 MSS=1460 SACK_PERM=1
1222	2019-10-22 17:49:03.379023	173.37.145.8	192.168.0.100	TCP	58		443 → 39924 [SYN, ACK] Seq=236460465 Ack=427175839 Win=8190 Len=0 MSS=1336
1223	2019-10-22 17:49:03.379298	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427175839 Ack=236460466 Win=29200 Len=0
1224	2019-10-22 17:49:03.380336	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
1225	2019-10-22 17:49:03.380732	173.37.145.8	192.168.0.100	TCP	54		443 → 39924 [ACK] Seq=236460466 Ack=427176356 Win=32768 Len=0
1226	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TLSv1.2	150		Server Hello
1227	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TCP	1384		443 → 39924 [PSH, ACK] Seq=236460562 Ack=427176356 Win=32768 Len=1330
1228	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TLSv1.2	155		Certificate
1229	2019-10-22 17:49:03.710107	173.37.145.8	192.168.0.100	TLSv1.2	63		Server Hello Done
1230	2019-10-22 17:49:03.710412	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236460562 Win=29200 Len=0
1231	2019-10-22 17:49:03.710519	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236461892 Win=31920 Len=0
1232	2019-10-22 17:49:03.710519	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236461993 Win=31920 Len=0
1233	2019-10-22 17:49:03.710534	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=427176356 Ack=236462002 Win=31920 Len=0
1234	2019-10-22 17:49:03.710626	192.168.0.100	173.37.145.8	TLSv1.2	61		Alert (Level: Fatal, Description: Unknown CA)
1235	2019-10-22 17:49:03.710641	173.37.145.8	192.168.0.100	TCP	54		443 → 39924 [ACK] Seq=236462002 Ack=427176363 Win=32768 Len=0
1236	2019-10-22 17:49:03.710748	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [RST, ACK] Seq=427176363 Ack=236462002 Win=31920 Len=0
1237	2019-10-22 17:49:03.710870	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [RST] Seq=427176363 Win=0 Len=0

```

Length: 1426
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1422
    Certificates Length: 1419
  Certificates (1419 bytes)
    Certificate Length: 1416
  Certificate: 308205843082046ca003020102020d00aa23af5d607e0000... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San
    signedCertificate
      version: v3 (2)
      serialNumber: 0x00aa23af5d607e00002f423880
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 3 items (id-at-commonName=FTD4100_MITM,id-at-organizationalUnitName=FTD_OU,id-at-organizationName=FTD_O)
          RDNSSequence item: 1 item (id-at-organizationName=FTD_O)
          RDNSSequence item: 1 item (id-at-organizationalUnitName=FTD_OU)
          RDNSSequence item: 1 item (id-at-commonName=FTD4100_MITM)
      validity
  
```

CAPO muestra:

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1169	2019-10-22 17:49:03.212849	192.168.0.100	173.37.145.8	TCP	78		39924 → 443 [SYN] Seq=623942018 Win=29200 Len=0 MSS=1380 SACK_PERM=1 TSval=
1170	2019-10-22 17:49:03.378962	173.37.145.8	192.168.0.100	TCP	62		443 → 39924 [SYN, ACK] Seq=4179450724 Ack=623942019 Win=8190 Len=0 MSS=1336
1171	2019-10-22 17:49:03.379329	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942019 Ack=4179450725 Win=29200 Len=0
1172	2019-10-22 17:49:03.380793	192.168.0.100	173.37.145.8	TLSv1.2	512	tools.cisco.com	Client Hello
1173	2019-10-22 17:49:03.545748	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179450725 Ack=623942473 Win=34780 Len=1330 [TCP
1174	2019-10-22 17:49:03.545809	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179452055 Ack=623942473 Win=34780 Len=1330 [TCP
1175	2019-10-22 17:49:03.545824	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942473 Ack=4179453385 Win=65535 Len=0
1176	2019-10-22 17:49:03.545915	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179453385 Ack=623942473 Win=34780 Len=1330 [TCP
1177	2019-10-22 17:49:03.545961	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4179454715 Ack=623942473 Win=34780 Len=1330 [TCP
1178	2019-10-22 17:49:03.545961	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942473 Ack=4179456045 Win=65535 Len=0
1179	2019-10-22 17:49:03.709420	173.37.145.8	192.168.0.100	TLSv1.2	82		Server Hello, Certificate, Server Hello Done
1180	2019-10-22 17:49:03.710687	192.168.0.100	173.37.145.8	TLSv1.2	65		Alert (Level: Fatal, Description: Unknown CA)
1181	2019-10-22 17:49:03.710885	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [FIN, PSH, ACK] Seq=623942480 Ack=4179456069 Win=65535 Len=0
1182	2019-10-22 17:49:03.874542	173.37.145.8	192.168.0.100	TCP	58		443 → 39924 [RST, ACK] Seq=4179456069 Ack=623942480 Win=9952 Len=0

```

Length: 5339
  Handshake Protocol: Server Hello
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 5240
    Certificates Length: 5237
  Certificates (5237 bytes)
    Certificate Length: 2025
  Certificate: 308207e5308205cda00302010202143000683b0f7504f7b2... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose,
    signedCertificate
      algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
      encrypted: 6921d084f7a6f6167058f14e2aad8b98b4e6c971ea6ea3b4...
    Certificate Length: 1736
  Certificate: 308206c4308204aca00302010202147517167783d0437eb5... (id-at-commonName=HydrantID SSL ICA G2,id-at-organizationName=HydrantID (Avalanche Cloud Corporation),id
    signedCertificate
      version: v3 (2)
      serialNumber: 0x7517167783d0437eb556c357946e4563b8ebd3ac
      signature (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 3 items (id-at-commonName=QuoVadis Root CA 2,id-at-organizationName=QuoVadis Limited,id-at-countryName=BM)
      validity
  
```

Estas capturas demuestran que el firewall de tránsito modifica el certificado de servidor (MITM)

Acción 2. Compruebe los registros del dispositivo.

Puede recopilar el paquete FMC TS como se describe en este documento:

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

En este caso, el archivo /dir-archives/var-log/process_stdout.log muestra mensajes como este:

```
<#root>
```

```
SOUT: 10-23 05:45:14 2019-10-23 05:45:36 s1a[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 s1a[10068]: *Wed .967 UTC: CH-LIB-TRACE: ch_pf_curl_is_cert_is
cert issue checking, ret 60, url "https://tools.cisco.com/its/
```

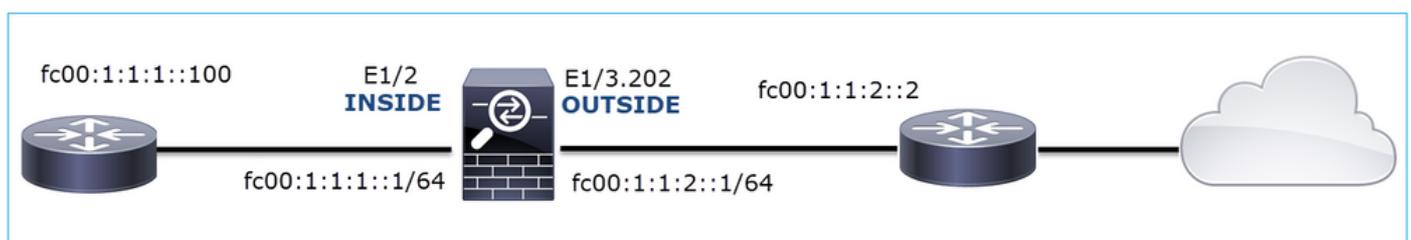
Solución recomendada

Inhabilite el MITM para el flujo específico de modo que FMC pueda registrarse correctamente en la nube de Smart Licensing.

Caso 11. Problema de conectividad IPv6

Descripción del problema: los hosts internos (situados detrás de la interfaz INTERNA del firewall) no pueden comunicarse con los hosts externos (hosts situados detrás de la interfaz EXTERNA del firewall).

Esta imagen muestra la topología:



Flujo afectado:

IP de origen: fc00:1:1:1::100

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

Protocolo: cualquiera

Análisis de captura

Activar capturas en el motor LINA de FTD.

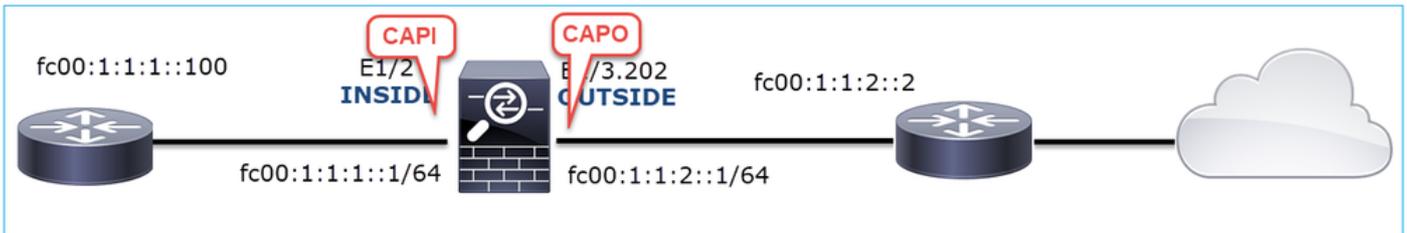
```
<#root>
```

```
firepower#
```

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

```
firepower#
```

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



Capturas: escenario no funcional

Estas capturas se realizaron en paralelo con una prueba de conectividad ICMP de IP fc00:1:1:1:100 (router interno) a IP fc00:1:1:2:2 (router ascendente).

La captura en la interfaz de firewall INSIDE contiene:

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:fc d8	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:fc d8	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:fc d8	ICMPv6	86	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fc d8 from 00:be:75:f6:1d:ae
13	2019-10-24 13:02:22.063000	fe80::4e4e:35ff:fefc:fc d8	fe80::2be:75ff:fef6:1dae	ICMPv6	78	Neighbor Advertisement fe80::4e4e:35ff:fefc:fc d8 (rtr, sol)

Puntos clave:

1. El router envía un mensaje de solicitud de vecino IPv6 y solicita la dirección MAC del dispositivo ascendente (IP fc00:1:1:1::1).
2. El firewall responde con un anuncio de vecino IPv6.
3. El router envía una solicitud de eco ICMP.
4. El firewall envía un mensaje de solicitud de vecino IPv6 y solicita la dirección MAC del dispositivo descendente (fc00:1:1:1::100).
5. El router responde con un anuncio de vecino IPv6.
6. El router envía solicitudes de eco ICMP IPv6 adicionales.

La captura en la interfaz EXTERNA del firewall contiene:

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	18	Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
4	2019-10-24 13:02:09.001815	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
5	2019-10-24 13:02:10.025938	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
6	2019-10-24 13:02:10.998132	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
7	2019-10-24 13:02:11.050015	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
8	2019-10-24 13:02:12.066082	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Solicitation for fe80::2be:75ff:fef6:1d8e from 4c:4e:35:fc:fc:d8
9	2019-10-24 13:02:12.066234	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	82	Neighbor Advertisement fe80::2be:75ff:fef6:1d8e (rtr, sol)
10	2019-10-24 13:02:12.998422	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
11	2019-10-24 13:02:13.002105	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
12	2019-10-24 13:02:14.090251	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
13	2019-10-24 13:02:14.998544	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
14	2019-10-24 13:02:15.178350	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
15	2019-10-24 13:02:17.059963	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	90	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8 from 00:be:75:f6:1d:8e
16	2019-10-24 13:02:17.062512	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	82	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

Puntos clave:

1. El firewall envía un mensaje de solicitud de vecino IPv6 que solicita la dirección MAC del dispositivo ascendente (IP fc00:1:1:2::2).
2. El router responde con un anuncio de vecino IPv6.
3. El firewall envía una solicitud de eco ICMP IPv6.
4. El dispositivo ascendente (router fc00:1:1:2:2) envía un mensaje de solicitud de vecino IPv6 que solicita la dirección MAC de la dirección IPv6 fc00:1:1:1::100.
5. El firewall envía una solicitud de eco ICMP IPv6 adicional.
6. El router ascendente envía un mensaje de solicitud de vecino IPv6 adicional que solicita la dirección MAC de la dirección IPv6 fc00:1:1:1::100.

El punto 4 es muy interesante. Normalmente, el router ascendente solicita la dirección MAC de la interfaz de firewall OUTSIDE (fc00:1:1:2::2), pero en su lugar, solicita la fc00:1:1:1::100. Esto es una indicación de un error de configuración.

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Compruebe la tabla de vecinos IPv6.

La tabla de vecinos IPv6 del firewall se ha rellenado correctamente.

```
<#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
```

Acción 2. Compruebe la configuración de IPv6.

Esta es la configuración del firewall.

```
<#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
```

La configuración del dispositivo ascendente revela el error de configuración:

<#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
```

Capturas - Escenario funcional

El cambio de máscara de subred (de /48 a /64) solucionó el problema. Esta es la captura CAPI en

el escenario funcional.

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)

Punto clave:

1. El router envía un mensaje de solicitud de vecino IPv6 que solicita la dirección MAC del dispositivo ascendente (IP fc00:1:1:1::1).
2. El firewall responde con un anuncio de vecino IPv6.
3. El router envía solicitudes de eco ICMP y obtiene respuestas de eco.

Contenido de CAPO:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 15:17:20.678645	fe80::2be:75ff:fe...	ff02::1:ff00:2	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
2	2019-10-24 15:17:20.681818	fc00:1:1:2::2	fe80::2be:75ff:fe...	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
3	2019-10-24 15:17:22.674342	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=1, hop limit=64 (reply in 6)
4	2019-10-24 15:17:22.677943	fc00:1:1:2::2	ff02::1:ff00:1	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::1 from 4c:4e:35:fc:fc:d8
5	2019-10-24 15:17:22.678096	fc00:1:1:2::1	fc00:1:1:2::2	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:8e
6	2019-10-24 15:17:22.678462	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=1, hop limit=64 (request in 3)
7	2019-10-24 15:17:24.674449	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=2, hop limit=64 (reply in 8)
8	2019-10-24 15:17:24.674785	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=2, hop limit=64 (request in 7)
9	2019-10-24 15:17:24.675395	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=3, hop limit=64 (reply in 10)
10	2019-10-24 15:17:24.675700	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=3, hop limit=64 (request in 9)
11	2019-10-24 15:17:24.676448	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=4, hop limit=64 (reply in 12)
12	2019-10-24 15:17:24.676738	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118	Echo (ping) reply id=0x097e, seq=4, hop limit=64 (request in 11)

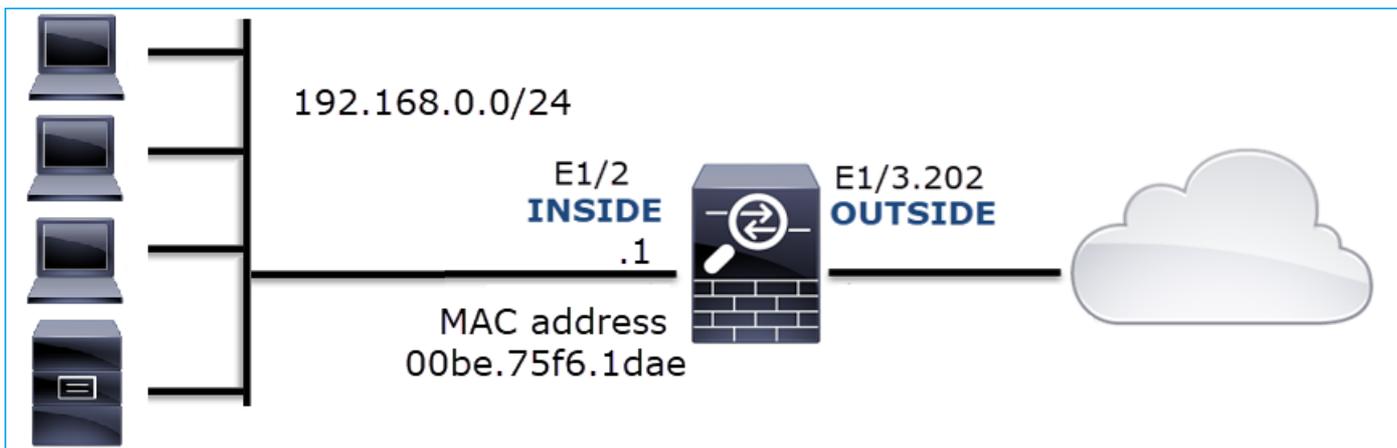
Puntos clave:

1. El firewall envía un mensaje de solicitud de vecino IPv6 que solicita la dirección MAC del dispositivo ascendente (IP fc00:1:1:2::2).
2. El firewall responde con un anuncio de vecino IPv6.
3. El firewall envía una solicitud de eco ICMP.
4. El router envía un mensaje de solicitud de vecino IPv6 que solicita la dirección MAC del dispositivo de flujo descendente (IP fc00:1:1:1::1).
5. El firewall responde con un anuncio de vecino IPv6.
6. El firewall envía solicitudes de eco ICMP y obtiene respuestas de eco.

Caso 12. Problema de conectividad intermitente (envenenamiento ARP)

Descripción del problema: los hosts internos (192.168.0.x/24) tienen problemas de conectividad intermitentes con los hosts de la misma subred

Esta imagen muestra la topología:



Flujo afectado:

IP de origen: 192.168.0.x/24

Dst IP: 192.168.0.x/24

Protocolo: cualquiera

La memoria caché ARP de un host interno parece estar contaminada:

```

C:\Windows\system32\cmd.exe
C:\Users\mzafeirol>arp -a

Interface: 192.168.0.55 --- 0xb
Internet Address      Physical Address      Type
192.168.0.1          00-be-75-f6-1d-ae    dynamic
192.168.0.22         00-be-75-f6-1d-ae    dynamic
192.168.0.23         00-be-75-f6-1d-ae    dynamic
192.168.0.24         00-be-75-f6-1d-ae    dynamic
192.168.0.25         00-be-75-f6-1d-ae    dynamic
192.168.0.26         00-be-75-f6-1d-ae    dynamic
192.168.0.27         00-be-75-f6-1d-ae    dynamic
192.168.0.28         00-be-75-f6-1d-ae    dynamic
192.168.0.29         00-be-75-f6-1d-ae    dynamic
192.168.0.30         00-be-75-f6-1d-ae    dynamic
192.168.0.88         00-be-75-f6-1d-ae    dynamic
192.168.0.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250     01-00-5e-7f-ff-fa    static

C:\Users\mzafeirol>

```

Análisis de captura

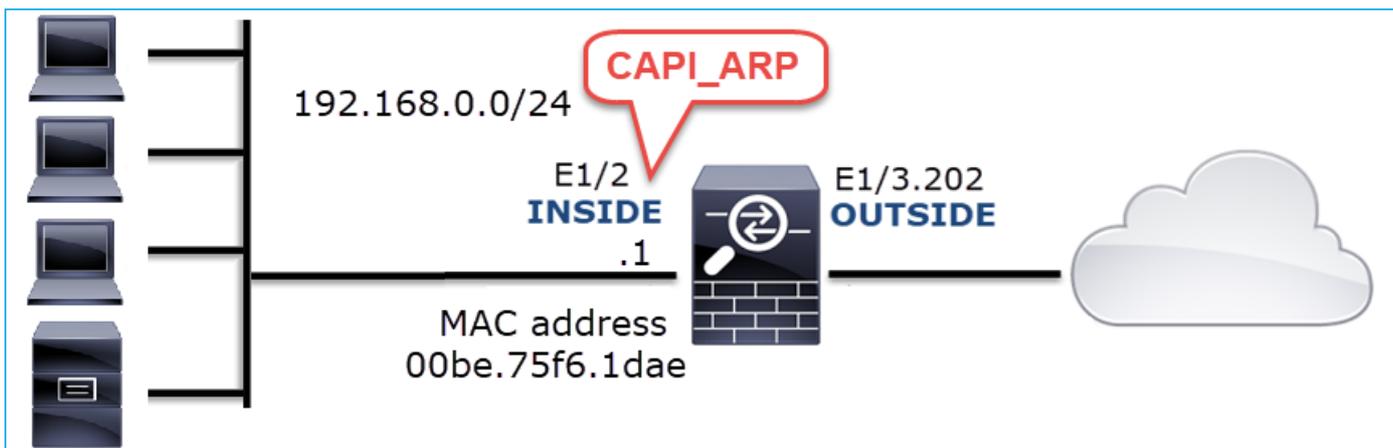
Habilitar una captura en el motor FTD LINA

Esta captura solo captura paquetes ARP en la interfaz INSIDE:

```
<#root>
```

```
firepower#
```

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



Capturas - Escenario no funcional:

La captura en la interfaz interna del firewall contiene.

No.	Time	Source	Destination	Protocol	Length	Info
4	2019-10-25 10:01:55.179571	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.23? Tell 192.168.0.55
5	2019-10-25 10:01:55.17969	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.23 is at 00:be:75:f6:1d:ae
35	2019-10-25 10:02:13.050397	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.24? Tell 192.168.0.55
36	2019-10-25 10:02:13.050488	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.24 is at 00:be:75:f6:1d:ae
47	2019-10-25 10:02:19.284683	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.25? Tell 192.168.0.55
48	2019-10-25 10:02:19.284775	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.25 is at 00:be:75:f6:1d:ae
61	2019-10-25 10:02:25.779821	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.26? Tell 192.168.0.55
62	2019-10-25 10:02:25.779912	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.26 is at 00:be:75:f6:1d:ae
76	2019-10-25 10:02:31.978175	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.27? Tell 192.168.0.55
77	2019-10-25 10:02:31.978251	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.27 is at 00:be:75:f6:1d:ae
97	2019-10-25 10:02:38.666515	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.28? Tell 192.168.0.55
98	2019-10-25 10:02:38.666606	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.28 is at 00:be:75:f6:1d:ae
121	2019-10-25 10:02:47.384074	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.29? Tell 192.168.0.55
122	2019-10-25 10:02:47.384150	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.29 is at 00:be:75:f6:1d:ae
137	2019-10-25 10:02:53.539995	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.30? Tell 192.168.0.55
138	2019-10-25 10:02:53.540087	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	2	42 192.168.0.30 is at 00:be:75:f6:1d:ae

Puntos clave:

1. El firewall recibe varias solicitudes ARP para IP dentro de la red 192.168.0.x/24
2. El firewall responde a todos ellos (proxy-ARP) con su propia dirección MAC

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Verifique la configuración de NAT.

Con respecto a la configuración de NAT, hay casos en los que la palabra clave no-proxy-arp puede evitar el comportamiento anterior:

```
<#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_4.4.4.4
no-proxy-arp
```

Acción 2. Inhabilite la funcionalidad proxy-arp en la interfaz de firewall.

Si la palabra clave 'no-proxy-arp' no resuelve el problema, intente inhabilitar el ARP proxy en la interfaz misma. En el caso de FTD, en el momento de escribir este documento, debe utilizar FlexConfig e implementar el comando (especifique el nombre de interfaz adecuado).

```
sysopt noproxyarp INSIDE
```

Caso 13. Identificar identificadores de objeto SNMP (OID) que provocan bloqueos de CPU

Este caso demuestra cómo ciertos OID de SNMP para el sondeo de memoria fueron identificados como la causa raíz de acaparamiento de CPU (problema de rendimiento) basado en el análisis de capturas de paquetes de SNMP versión 3 (SNMPv3).

Descripción del problema: Los desbordamientos en las interfaces de datos aumentan continuamente. Investigaciones posteriores revelaron que también hay acaparamientos de CPU (causados por el proceso SNMP) que son la causa raíz de los desbordamientos de la interfaz.

El siguiente paso en el proceso de solución de problemas fue identificar la causa raíz de los acaparamientos de CPU causados por el proceso SNMP y, en particular, reducir el alcance del problema para identificar los identificadores de objeto SNMP (OID) que, cuando se sondea, podrían potencialmente dar lugar a acaparamientos de CPU.

Actualmente, el motor FTD LINA no proporciona un comando 'show' para los OID SNMP que se sondean en tiempo real.

La lista de OIDs SNMP para sondeo se puede recuperar desde la herramienta de monitoreo SNMP, sin embargo, en este caso, hubo estos factores preventivos:

- El administrador de FTD no tenía acceso a la herramienta de supervisión SNMP
- SNMP versión 3 con autenticación y cifrado de datos para la privacidad se configuró en FTD

Análisis de captura

Dado que el administrador de FTD tenía las credenciales para la autenticación SNMP versión 3 y

el cifrado de datos, se propuso este plan de acción:

1. Captura de paquetes SNMP
2. Guarde las capturas y utilice las preferencias del protocolo SNMP de Wireshark para especificar las credenciales de SNMP versión 3 para descifrar los paquetes de SNMP versión 3. Las capturas descifradas se utilizan para el análisis y la recuperación de OID de SNMP

Configure las capturas de paquetes SNMP en la interfaz que se utiliza en la configuración de host del servidor SNMP:

```
<#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 capsntp interface management buffer 10000000 match udp host 192.168.10.10 host 192.168.5.254 eq
```

```
firepower#
```

```
show capture capsntp
```

```
capture capsntp type raw-data buffer 10000000 interface outside [Capturing -
```

```
9512
```

```
bytes]
```

```
match udp host 192.168.10.10 host 192.168.5.254 eq snmp
```

No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	encryptedPDU: privKey Unknown
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	encryptedPDU: privKey Unknown
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	encryptedPDU: privKey Unknown
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	encryptedPDU: privKey Unknown
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	encryptedPDU: privKey Unknown
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	encryptedPDU: privKey Unknown
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	encryptedPDU: privKey Unknown
15	1.317	SNMP	192.168.10.10	65484	161	192.168.5.254	205	encryptedPDU: privKey Unknown
16	1.318	SNMP	192.168.5.254	161	65484	192.168.10.10	513	encryptedPDU: privKey Unknown
17	17.595	SNMP	192.168.10.10	62008	161	192.168.5.254	100	getBulkRequest
18	17.595	SNMP	192.168.5.254	161	62008	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
19	17.749	SNMP	192.168.10.10	62008	161	192.168.5.254	197	encryptedPDU: privKey Unknown
20	17.749	SNMP	192.168.5.254	161	62008	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
21	17.898	SNMP	192.168.10.10	62008	161	192.168.5.254	199	encryptedPDU: privKey Unknown
22	17.899	SNMP	192.168.5.254	161	62008	192.168.10.10	678	encryptedPDU: privKey Unknown
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	encryptedPDU: privKey Unknown
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	encryptedPDU: privKey Unknown
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	encryptedPDU: privKey Unknown

```

<[Destination Host: 192.168.5.254]>
<[Source or Destination Host: 192.168.5.254]>
> User Datagram Protocol, Src Port: 65484, Dst Port: 161
  Simple Network Management Protocol
    msgVersion: snmpv3 (3)
    > msgGlobalData
    > msgAuthoritativeEngineID: 80000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40_
    msgAuthoritativeEngineBoots: 0
    msgAuthoritativeEngineTime: 0
    msgUserName: netmonv3
    msgAuthenticationParameters: ff5176f5973c30b62ffc11b8
    msgPrivacyParameters: 000040e100003196
    > msgData: encryptedPDU (1)
      encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703_

```

Puntos clave:

1. Direcciones/puertos de origen y destino SNMP.
2. No se pudo descodificar la PDU del protocolo SNMP porque Wireshark desconoce privKey.
3. El valor de la primitiva de la PDU cifrada.

Acciones recomendadas

Las acciones enumeradas en esta sección tienen como objetivo reducir aún más el problema.

Acción 1. Descifre las capturas SNMP.

Guarde las capturas y edite las preferencias del protocolo SNMP de Wireshark para especificar las credenciales de SNMP versión 3 para descifrar los paquetes.

```
<#root>
```

```
firepower#
```

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

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

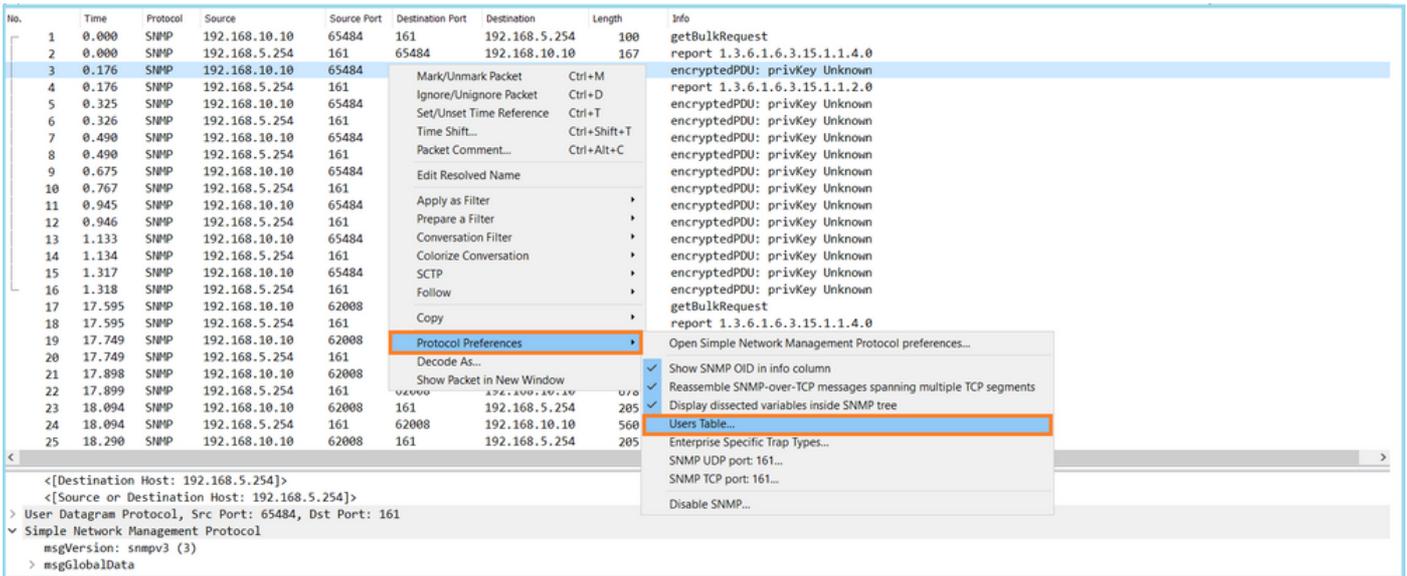
```
Address or name of remote host []? 192.168.10.253
```

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

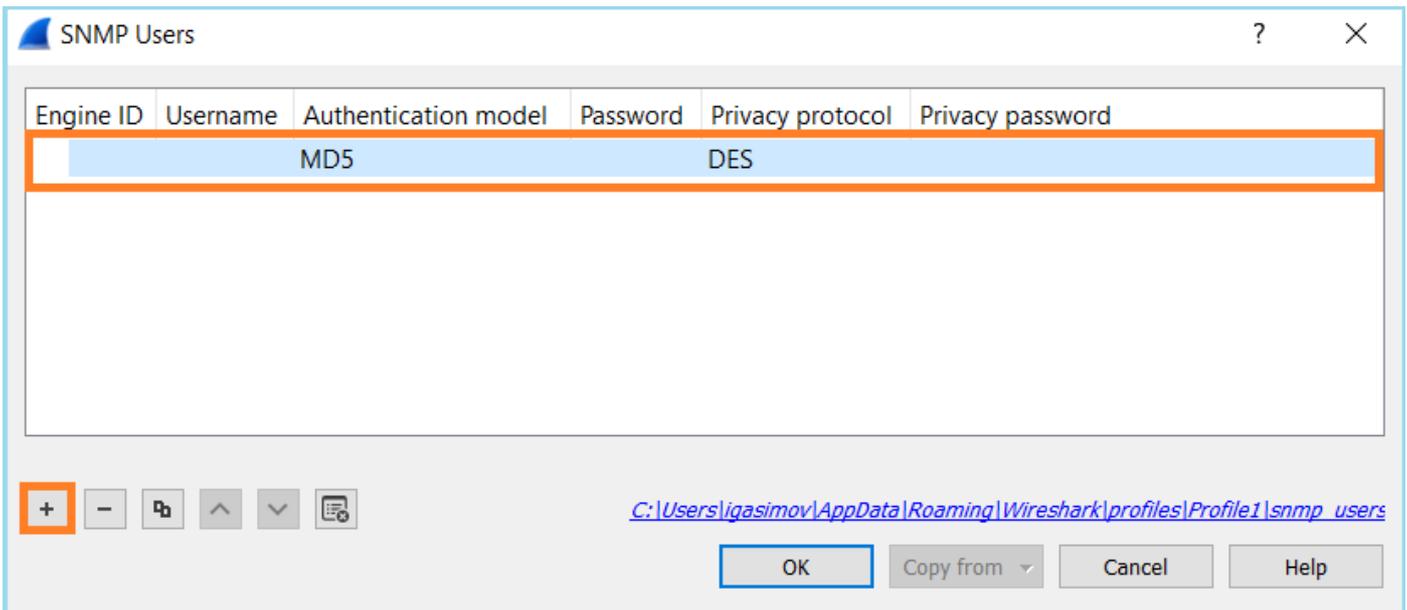
```
!!!!!!
```

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

Abra el archivo de captura en Wireshark, seleccione un paquete SNMP y navegue hasta Protocol Preferences > Users Table, como se muestra en la imagen:



En la tabla de usuarios SNMP se especificaron el nombre de usuario, el modelo de autenticación, la contraseña de autenticación, el protocolo de privacidad y la contraseña de privacidad de la versión 3 de SNMP (las credenciales reales no se muestran a continuación):



Una vez que se aplicó la configuración de usuarios SNMP, Wireshark mostró PDU de SNMP descifradas:

No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	getBulkRequest 1.3.6.1.4.1.9.9.221.1
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	getBulkRequest 1.3.6.1.4.1.9.9.221.1
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.7.1.1
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response 1.3.6.1.4.1.9.9.221.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.7.1.2 1.3.6.1.4.1.9.9.221.1.1.1.8.1.8
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.18.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response 1.3.6.1.4.1.9.9.221.1.1.1.17.1.1 1.3.6.1.4.1.9.9.221.1.1.1.17.1.2 1.3.6.1.4.1.9.9.221.1.1.1.19.1.2 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.392.1.1.1.0
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	get-response 1.3.6.1.4.1.9.9.221.1.1.1.392.1.1.1.0 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 1.3.6.1.4.1.9.9.221.1.1.1.18.1.8
15	1.317	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8
16	1.318	SNMP	192.168.5.254	161	65484	192.168.10.10	513	get-response 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
17	17.595	SNMP	192.168.10.10	62008	161	192.168.5.254	100	getBulkRequest
18	17.595	SNMP	192.168.5.254	161	62008	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
19	17.749	SNMP	192.168.10.10	62008	161	192.168.5.254	197	getBulkRequest 1.3.6.1.4.1.9.9.221.1
20	17.749	SNMP	192.168.5.254	161	62008	192.168.10.10	192	report 1.3.6.1.6.3.15.1.1.2.0
21	17.898	SNMP	192.168.10.10	62008	161	192.168.5.254	199	getBulkRequest 1.3.6.1.4.1.9.9.221.1
22	17.899	SNMP	192.168.5.254	161	62008	192.168.10.10	678	get-response 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.2.1.2 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	get-response 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.5.1.2 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8


```

msgData: encryptedPDU (1)
  encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703...
    Decrypted ScopedPDU: 303b04198000009fe1c6dad4930a00ef1fec2301621a415...
      contextEngineID: 8000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40...
      contextName:
      data: getBulkRequest (5)
        getBulkRequest
          request-id: 5620
          non-repeaters: 0
          max-repetitions: 16
          variable-bindings: 1 item
            1.3.6.1.4.1.9.9.221.1: Value (Null)
              Object Name: 1.3.6.1.4.1.9.9.221.1 (iso.3.6.1.4.1.9.9.221.1)
              Value (Null)
  
```

Puntos clave:

1. Las herramientas de monitoreo SNMP usaron SNMP getBulkRequest para consultar y recorrer el OID primario 1.3.6.1.4.1.9.9.221.1 y los OID relacionados.
2. El FTD respondió a cada getBulkRequest con get-response que contiene OID relacionados con 1.3.6.1.4.1.9.9.221.1.

Acción 2. Identifique los OID de SNMP.

[SNMP Object Navigator](#) mostró que OID 1.3.6.1.4.1.9.9.221.1 pertenece a la base de información de administración (MIB) denominada CISCO-ENHANCED-MEMPOOL-MIB, como se muestra en la imagen:

Tools & Resources
SNMP Object Navigator

HOME | SUPPORT | TOOLS & RESOURCES | **SNMP Object Navigator**

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

Translate | Browse The Object Tree

Related Tools
[Support Case Manager](#)
[Cisco Community](#)
[MIB Locator](#)

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

Enter OID or object name: 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)
|

```

Para mostrar los OID en formato legible para humanos en Wireshark:

1. Descargue MIB CISCO-ENHANCED-MEMPOOL-MIB y sus dependencias, como se muestra en la imagen:

Tools & Resources
SNMP Object Navigator

HOME | SUPPORT | TOOLS & RESOURCES | **SNMP Object Navigator**

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

Related Tools
[Support Case Manager](#)
[Cisco Community](#)
[MIB Locator](#)

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

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

Tools & Resources
SNMP Object Navigator

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

Support Case Manager
 Cisco Community
 MIB Locator

CISCO-ENHANCED-MEMPOOL-MIB

View compiling dependencies for other MIBS by [clearing](#) the page and selecting another MIB.

Compile the MIB

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

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

If you are using Internet Explorer click [here](#).

MIB Name	Version 1	Version 2	Dependencies
1. SNMPv2-SMI	Download	Download	View Dependencies
2. SNMPv2-TC	Download	Download	View Dependencies
3. SNMPv2-CONF	Not Required	Download	View Dependencies
4. SNMP-FRAMEWORK-MIB	Download	Download	View Dependencies
5. CISCO-SMI	Download	Download	View Dependencies
6. ENTITY-MIB	Download	Download	View Dependencies
7. HCNUM-TC	Download	Download	View Dependencies
8. RFC1155-SMI	Non-Cisco MIB	Non-Cisco MIB	-
9. RFC-1212	Non-Cisco MIB	Non-Cisco MIB	-
10. RFC-1215	Non-Cisco MIB	Non-Cisco MIB	-
11. SNMPv2-TC-v1	Non-Cisco MIB	Non-Cisco MIB	-
12. CISCO-ENHANCED-MEMPOOL-MIB	Download	Download	

2. En Wireshark en Edit > Preferences > Name Resolution ventana, la opción Enable OID Resolution está marcada. En la ventana SMI (MIB and PIB paths) especifique la carpeta con los MIB descargados y en SMI (MIB and PIB modules). CISCO-ENHANCED-MEMPOOL-MIB se agrega automáticamente a la lista de módulos:

The screenshot shows the Wireshark interface with the Name Resolution preferences window open. The 'Enable OID resolution' checkbox is checked. The SMI Paths window shows the directory path 'C:/Users/Administrator/Downloads/SNMPMIBS' selected. The SMI Modules window shows 'CISCO-ENHANCED-MEMPOOL-MIB' listed among the modules.

3. Una vez que se reinicia Wireshark, se activa la resolución OID:

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::compMIBObjects
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::compMIBObjects
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolType.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolType
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolAlternate.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPoc
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolValid.1.8
10	0.675	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsed.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsed
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolFree.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolUsedOvrflw.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemPc
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolHCUsed.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	600	get-response CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolFreeQue.1.1 CISCO-ENHANCED-MEMPOOL-MIB::compMemP


```

✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.1 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.1): System memory
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.1 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.1)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: System memory
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.2 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.2): System memory
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.2 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.2)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: System memory
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.3 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.3): MEMPOOL_MSGLYR
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.3 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.3)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_MSGLYR
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.4 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.4): MEMPOOL_HEAPCACHE_1
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.4 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.4)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_HEAPCACHE_1
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.5 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.5): MEMPOOL_HEAPCACHE_0
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.5 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.5)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_HEAPCACHE_0
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.6 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.6): MEMPOOL_DMA_ALT1
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.6 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.6)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_DMA_ALT1
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.7 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.7): MEMPOOL_DMA
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.7 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.7)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_DMA
✓ CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.8 (1.3.6.1.4.1.9.9.221.1.1.1.3.1.8): MEMPOOL_GLOBAL_SHARED
  Object Name: 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8 (CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName.1.8)
  CISCO-ENHANCED-MEMPOOL-MIB::compMemPoolName: MEMPOOL_GLOBAL_SHARED

```

Basándose en la salida descifrada del archivo de captura, la herramienta de supervisión SNMP consultaba periódicamente (con un intervalo de 10 segundos) los datos sobre la utilización de los grupos de memoria en el FTD. Como se explicó en el artículo de TechNote [ASA SNMP Polling for Memory-Related Statistics](#), el sondeo de la utilización de Global Shared Pool (GSP) con SNMP da como resultado un uso elevado de la CPU. En este caso de las capturas, estaba claro que la utilización del conjunto compartido global se sondeaba periódicamente como parte de la primitiva getBulkRequest de SNMP.

Para minimizar los acaparamientos de CPU causados por el proceso SNMP, se recomendó seguir los pasos de mitigación para los acaparadores de CPU para SNMP mencionados en el artículo y evitar sondear los OID relacionados con GSP. Sin el sondeo SNMP para los OIDs que se relacionan con el GSP, no se observaron acaparamientos de CPU causados por el proceso SNMP y la tasa de desbordamientos disminuyó significativamente.

Información Relacionada

- [Guías de configuración de Cisco Firepower Management Center](#)
- [Aclaración de acciones de reglas de políticas de control de acceso de defensa contra amenazas de Firepower](#)
- [Trabaje con capturas de Firepower Threat Defence y Packet Tracer](#)
- [Aprender Wireshark](#)

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).