

Analisi delle acquisizioni di Firepower Firewall per la risoluzione dei problemi di rete

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Come raccogliere ed esportare le clip sulla famiglia di prodotti NGFW?](#)

[Raccogli acquisizioni FXOS](#)

[Abilita e raccogli acquisizioni di linea FTD](#)

[Abilita e raccogli acquisizioni di snort FTD](#)

[Risoluzione dei problemi](#)

[Caso 1. Nessun TCP SYN su interfaccia in uscita](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Sintetico delle possibili cause e delle azioni consigliate](#)

[Caso 2. TCP SYN da client, TCP RST da server](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Caso 3. Handshake TCP a 3 vie + RST da un endpoint](#)

[Analisi acquisizione](#)

[3.1 - Handshake TCP a 3 vie + RST ritardato dal client](#)

[Azioni consigliate](#)

[3.2 - Handshake TCP a 3 vie + FIN/ACK ritardato dal client + RST ritardato dal server](#)

[Azioni consigliate](#)

[3.3 - Handshake TCP a 3 vie + RST ritardato dal client](#)

[Azioni consigliate](#)

[3.4 - Handshake TCP a 3 vie + RST immediato dal server](#)

[Azioni consigliate](#)

[Caso 4. TCP RST dal client](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Caso 5. Trasferimento TCP lento \(scenario 1\)](#)

[Scenario 1. Trasferimento lento](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Scenario 2. Trasferimento rapido](#)

[Caso 6. Trasferimento TCP lento \(scenario 2\)](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Esportare la cattura controllare la differenza di tempo tra i pacchetti in entrata e in uscita](#)

[Caso 7. Problema di connettività TCP \(pacchetti danneggiati\)](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Caso 8. Problema di connettività UDP \(pacchetti mancanti\)](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Caso 10. Problema di connettività HTTPS \(scenario 2\)](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Caso 12. Problema di connettività intermittente \(avvelenamento ARP\)](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Caso 13. Identificazione degli identificatori di oggetti \(OID\) SNMP che causano il blocco della CPU](#)

[Analisi acquisizione](#)

[Azioni consigliate](#)

[Informazioni correlate](#)

Introduzione

Questo documento descrive varie tecniche di analisi dell'acquisizione dei pacchetti che mirano a risolvere in modo efficace i problemi relativi alla rete.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Architettura della piattaforma Firepower
- Log NGFW
- Packet-tracer NGFW

Inoltre, prima di iniziare ad analizzare le acquisizioni dei pacchetti, si consiglia di soddisfare i seguenti requisiti:

- Conoscere il protocollo - Non iniziare a controllare l'acquisizione di un pacchetto se non si comprende come funziona il protocollo acquisito.
- Conoscere la topologia: è necessario conoscere tutti i dispositivi di transito. Se ciò non è possibile, è necessario conoscere almeno i dispositivi a monte e a valle.
- Conoscere l'accessorio - È necessario sapere in che modo il dispositivo gestisce i pacchetti, quali sono le interfacce interessate (in entrata e in uscita), qual è l'architettura del dispositivo e quali sono i vari punti di acquisizione.
- Conoscere la configurazione - È necessario sapere in che modo il dispositivo deve gestire il flusso di un pacchetto in termini di:
 - Interfaccia di routing/uscita

- Criteri applicati
- NAT (Network Address Translation)
- Conoscere gli strumenti disponibili - Oltre alle clip, si consiglia di essere pronti ad applicare altri strumenti e tecniche (come il logging e i tracer) e, se necessario, di correlarli ai pacchetti acquisiti.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- La maggior parte degli scenari si basa su FP4140 con software FTD 6.5.x.
- FMC con software 6.5.x.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

L'acquisizione dei pacchetti è uno degli strumenti di risoluzione dei problemi più sottovalutati oggi disponibili. Ogni giorno, Cisco TAC risolve molti problemi relativi all'analisi dei dati acquisiti.

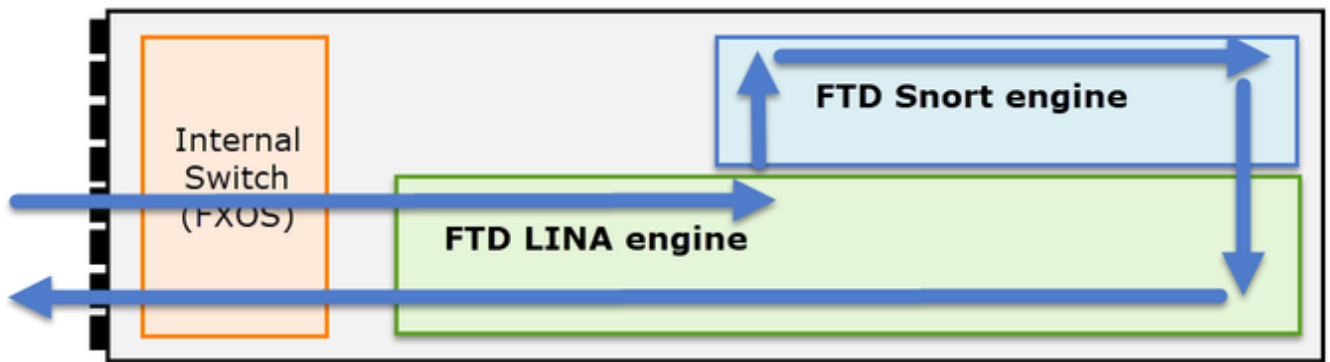
Obiettivo di questo documento è aiutare i tecnici di rete e della sicurezza a identificare e risolvere i problemi comuni della rete, basandosi principalmente sull'analisi dell'acquisizione dei pacchetti.

Tutti gli scenari presentati in questo documento si basano sui casi effettivi degli utenti rilevati nel Cisco Technical Assistance Center (TAC).

Il documento descrive i pacchetti acquisiti da un punto di vista di Cisco Next-Generation Firewall (NGFW), ma gli stessi concetti possono essere applicati anche ad altri tipi di dispositivi.

Come raccogliere ed esportare le clip sulla famiglia di prodotti NGFW?

Nel caso di un'appliance Firepower (1xxx, 21xx, 41xx, 93xx) e di un'applicazione Firepower Threat Defense (FTD), è possibile visualizzare un'elaborazione dei pacchetti come mostrato nell'immagine.



1. Un pacchetto entra nell'interfaccia in entrata e viene gestito dallo switch interno dello chassis.
2. Il pacchetto entra nel motore FTD Lina che esegue principalmente controlli L3/L4.
3. Se la politica richiede che il pacchetto sia ispezionato dal motore Snort (principalmente l'ispezione L7).
4. Il motore Snort restituisce un verdetto per il pacchetto.
5. In base a questo verdetto, il motore LINA elimina il pacchetto o lo inoltra.
6. Il pacchetto attraversa lo chassis attraverso lo switch interno dello chassis.

In base all'architettura mostrata, le clip FTD possono essere acquisite in tre (3) posti diversi:

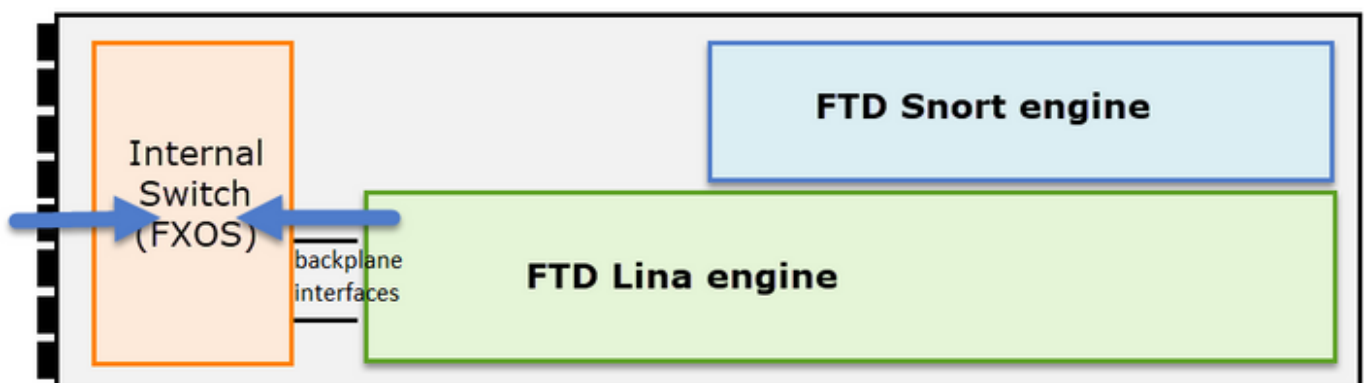
- FXOS
- Motore FTD Lina
- FTD Motore Snort

Raccogli acquisizioni FXOS

Il processo è descritto nel presente documento:

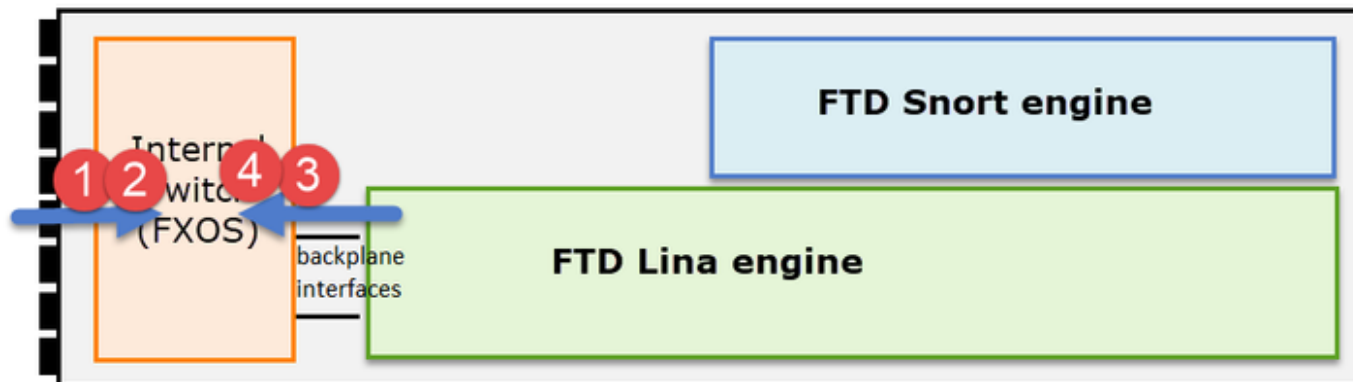
https://www.cisco.com/c/en/us/td/docs/security/firepower/pxos/pxos271/web-guide/b_GUI_FXOS_ConfigGuide_271/troubleshooting.html#concept_E8823CC63C934A909BBC0DF12F

Le riprese FXOS possono essere effettuate solo in entrata dal punto di vista dello switch interno; l'immagine mostra questa schermata.




Qui mostrati, questi sono due punti di acquisizione per direzione (a causa dell'architettura dello

switch interno).



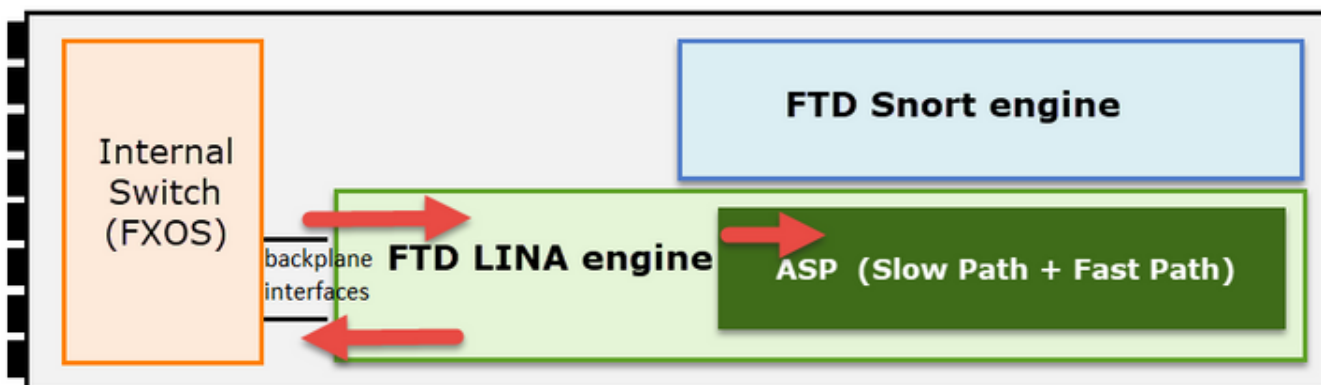
I pacchetti acquisiti ai punti 2, 3 e 4 hanno un tag di rete virtuale (VNTag).

 Nota: le acquisizioni a livello di chassis FXOS sono disponibili solo sulle piattaforme FP41xx e FP93xx. FP1xxx e FP21xx non offrono questa funzionalità.

Abilita e raccogli acquisizioni di linea FTD

Punti di acquisizione principali:

- Interfaccia in ingresso
- Interfaccia in uscita
- ASP (Accelerated Security Path)



È possibile utilizzare l'interfaccia utente di Firepower Management Center (FMC UI) o l'interfaccia CLI di FTD per abilitare e raccogliere le acquisizioni di Lina FTD.

Abilitare l'acquisizione dalla CLI sull'interfaccia INSIDE:

```
<#root>
```

```
firepower#
```

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

Questa acquisizione corrisponde al traffico tra gli indirizzi IP 192.168.103.1 e 192.168.101.1 in entrambe le direzioni.

Abilitare l'acquisizione ASP per visualizzare tutti i pacchetti scartati dal motore Lina FTD:

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

Esportare un'acquisizione Lina FTD in un server FTP:

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

Esportare un'acquisizione Lina FTD in un server TFTP:

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

A partire dalla versione 6.2.x di FMC è possibile abilitare e raccogliere le acquisizioni di Lina FTD dall'interfaccia utente di FMC.

Un altro modo per raccogliere le acquisizioni FTD da un firewall gestito da FMC è questo.

Passaggio 1

In caso di acquisizione LINA o ASP, copiare l'acquisizione sul disco FTD.

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

Source capture name [capin]?

Destination filename [capin.pcap]?

!!!!

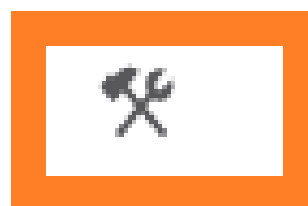
Passaggio 2

Passare alla modalità Expert, individuare l'acquisizione salvata e copiarla nella posizione /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
```

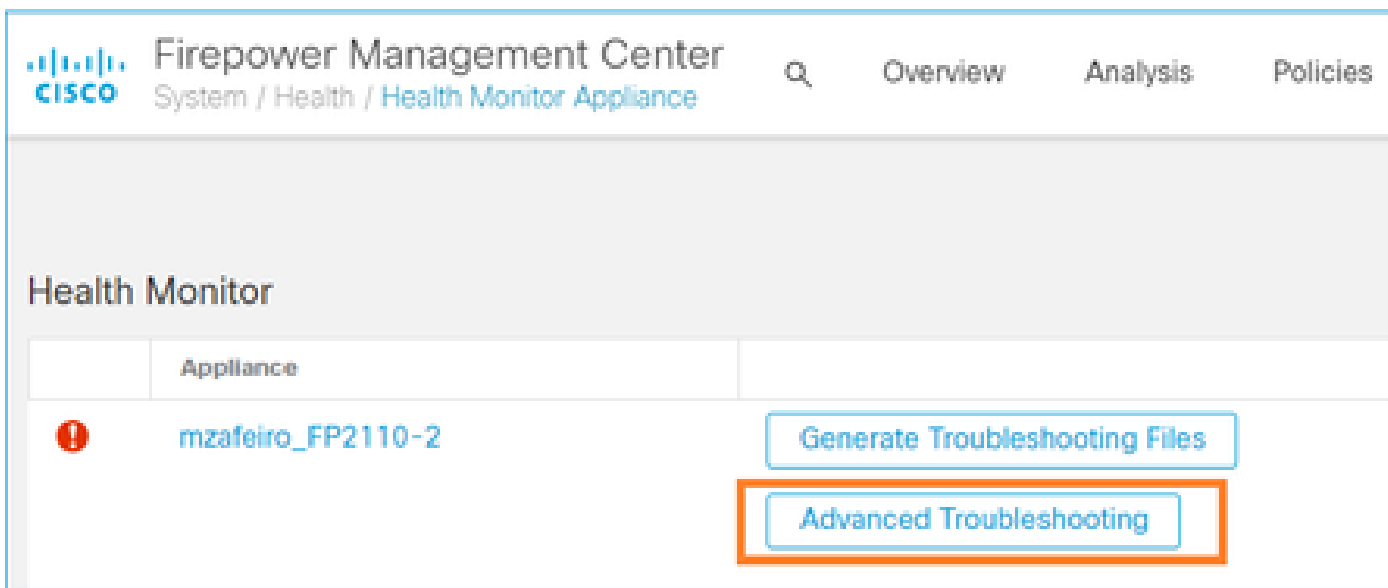
Passaggio 3

Accedere al FMC che gestisce l'FTD e selezionare Dispositivi > Gestione dispositivi. Individuare il dispositivo FTD e selezionare l'icona Risoluzione problemi:

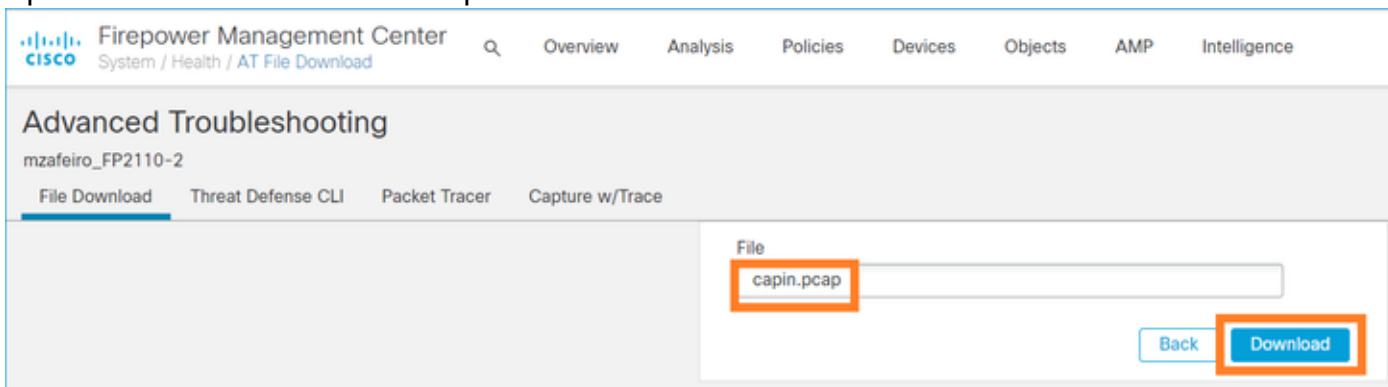


Passaggio 4

Selezionare Advanced Troubleshooting (Risoluzione avanzata problemi):



Specificare il nome del file di acquisizione e selezionare Download:

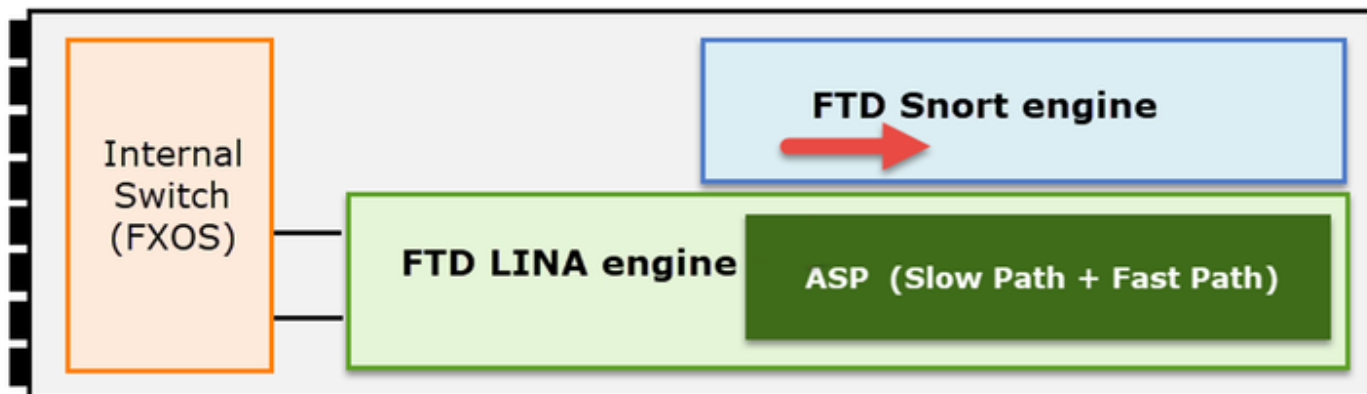


Per ulteriori esempi su come abilitare/raccogliere le acquisizioni dall'interfaccia utente di FMC, vedere questo documento:

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

Abilita e raccogli acquisizioni di snort FTD

Il punto di cattura è mostrato nell'immagine qui.



Abilita acquisizione a livello di 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
```

Per scrivere l'acquisizione in un file denominato capture.pcap e copiarlo tramite FTP su un server 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.

>

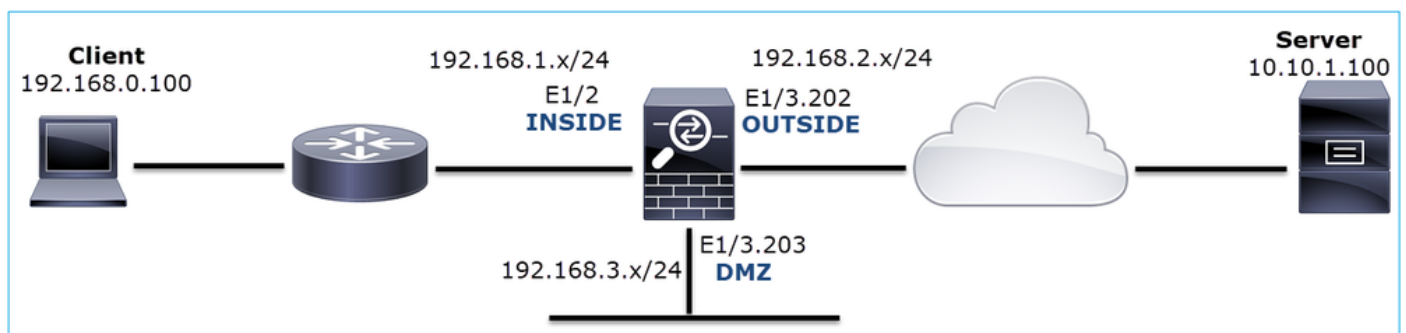
Per altri esempi di acquisizione a livello di snort che includono filtri di acquisizione diversi, consultare questo documento:

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

Risoluzione dei problemi

Caso 1. Nessun TCP SYN su interfaccia in uscita

La topologia è mostrata nell'immagine qui:



Descrizione del problema: HTTP non funziona

Flusso interessato:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocollo: TCP 80

Analisi acquisizione

Abilita le clip sul motore LINA FTD:

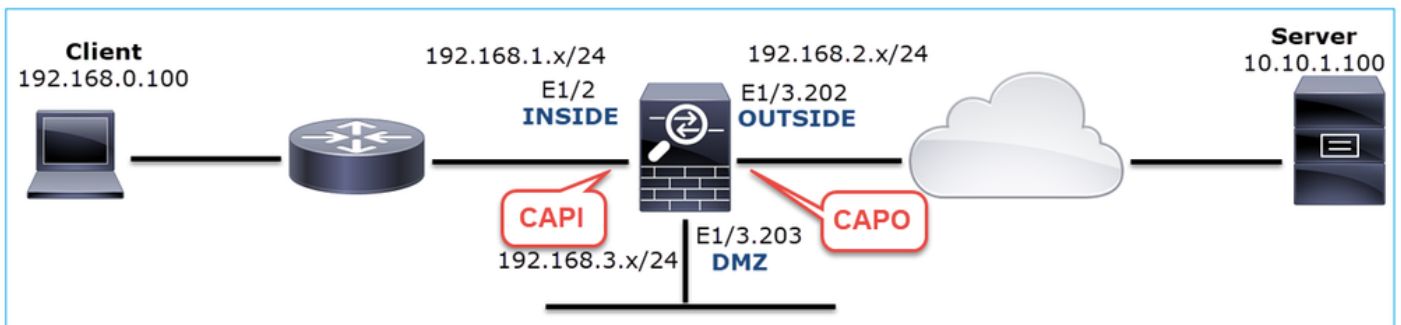
```
<#root>
```

```
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
```



Clip - Scenario funzionale:

Come base, è sempre molto utile avere acquisizioni da uno scenario funzionale.

La cattura è stata effettuata con l'interfaccia NGFW INSIDE, come mostrato nell'immagine:

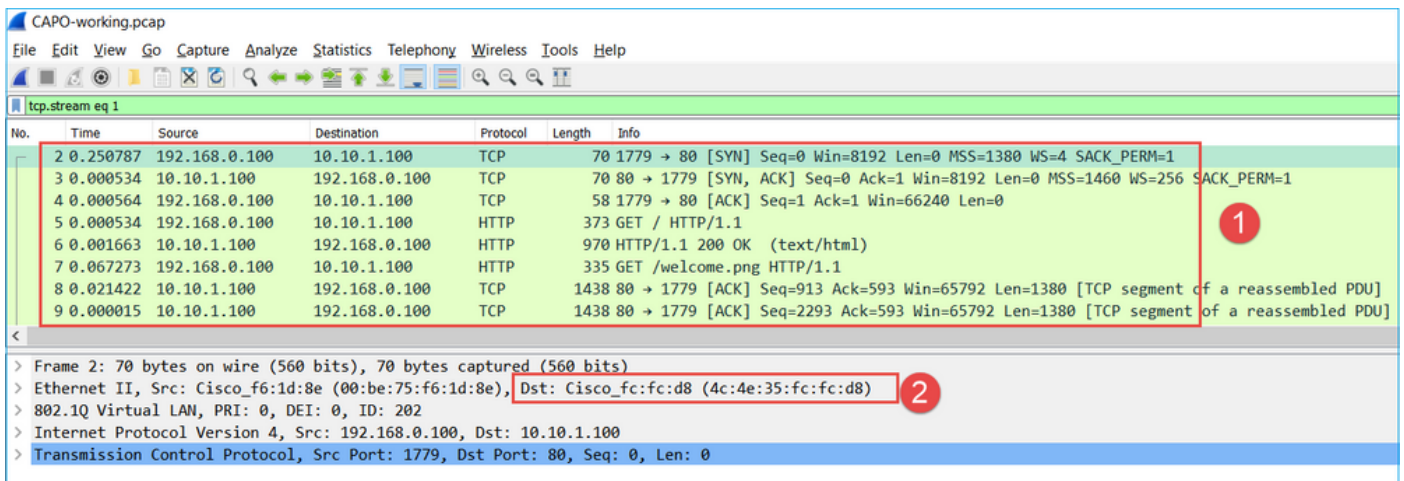
No.	Time	Source	Destination	Protocol	Length	Info
2	0.250878	192.168.0.100	10.10.1.100	TCP	66	1779 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	0.001221	10.10.1.100	192.168.0.100	TCP	66	80 → 1779 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
4	0.000488	192.168.0.100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

> Frame 2: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
> Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
> Transmission Control Protocol, Src Port: 1779, Dst Port: 80, Seq: 0, Len: 0

Considerazioni principali:

1. Handshake TCP a 3 vie.
2. Scambio bidirezionale di dati.
3. Nessun ritardo tra i pacchetti (in base alla differenza di tempo tra i pacchetti).
4. L'indirizzo MAC di origine è il dispositivo downstream corretto.

Le immagini acquisite tramite l'interfaccia NGFW OUTSIDE sono mostrate nell'immagine seguente:



Considerazioni principali:

1. Stessi dati dell'acquisizione CAPI.
2. L'indirizzo MAC di destinazione è il dispositivo upstream corretto.

Acquisizioni - Scenario non funzionale

Dalla CLI del dispositivo le clip hanno questo aspetto:

```
<#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
```

Contenuto 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

Questa è l'immagine di CAPI capture in 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	0.250420	192.168.0.100	10.10.1.100	TCP	66	3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	2.745781	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4	0.255074	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
5	5.751883	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
6	0.250070	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1

3 Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 > Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
 > Transmission Control Protocol, Src Port: 3171, Dst Port: 80, Seq: 0, Len: 0

Considerazioni principali:

1. Vengono visualizzati solo i pacchetti TCP SYN (nessun handshake a 3 vie TCP).

2. Impossibile stabilire due sessioni TCP (porta di origine 3171 e 3172). Il client di origine invia nuovamente i pacchetti TCP SYN. Questi pacchetti ritrasmessi vengono identificati da Wireshark come ritrasmissioni TCP.
3. Le ritrasmissioni TCP avvengono ogni ~3, 6, ecc.
4. L'indirizzo MAC di origine proviene dal dispositivo downstream corretto.

Sulla base delle due catture si può concludere che:

- Un pacchetto di una 5-tupla specifica (src/dst IP, src/dst port, protocollo) arriva sul firewall sull'interfaccia prevista (INSIDE).
- Un pacchetto non esce dal firewall sull'interfaccia prevista (ESTERNA).

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. controllare la traccia di un pacchetto emulato.

Utilizzare lo strumento packet-tracer per verificare come deve essere gestito un pacchetto dal firewall. Se il pacchetto viene scartato dai criteri di accesso del firewall, la traccia del pacchetto emulato avrà un aspetto simile al seguente output:

```
<#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

Azione 2. Controllare le tracce dei pacchetti live.

Abilitare la traccia dei pacchetti per controllare come i pacchetti TCP SYN reali vengono gestiti dal firewall. Per impostazione predefinita, vengono tracciati solo i primi 50 pacchetti in entrata:

```
<#root>
```

```
firepower#
```

```
capture CAPI trace
```

Cancellare il buffer di acquisizione:

```
<#root>
```

```
firepower#
```

```
clear capture /all
```

Se il pacchetto viene scartato dai criteri di accesso del firewall, la traccia avrà un aspetto simile al seguente output:

```
<#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
```

Azione 3. Controllare i registri Lina FTD.

Per configurare Syslog su FTD tramite FMC, controllare questo documento:

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

Si consiglia di configurare un server Syslog esterno per i log Lina FTD. Se non è configurato alcun server Syslog remoto, abilitare i log del buffer locale sul firewall durante la risoluzione dei problemi. La configurazione del log mostrata in questo esempio è un buon punto di partenza:

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

Impostare il cercapersone terminale su 24 linee per controllare il cercapersone terminale:

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

Cancellare il buffer di acquisizione:

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

Verificare la connessione e controllare i registri con un filtro parser. Nell'esempio i pacchetti vengono scartati dai criteri di accesso del firewall:

```
<#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
```

Azione 4. Controllare le gocce ASP del firewall.

Se si sospetta che il pacchetto sia stato scartato dal firewall, è possibile visualizzare i contatori di tutti i pacchetti scartati dal firewall a livello di 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
```

È possibile abilitare le acquisizioni per visualizzare tutte le perdite a livello di software ASP:

```
<#root>
```

```
firepower#
```

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

 Suggestivo: se non si è interessati al contenuto del pacchetto, è possibile acquisire solo le intestazioni del pacchetto (opzione solo intestazioni). In questo modo è possibile acquisire un numero molto maggiore di pacchetti nel buffer di acquisizione. Inoltre, è possibile aumentare la dimensione del buffer di acquisizione (per impostazione predefinita è di 500 Kbyte) fino a un valore di 32 Mbyte (opzione del buffer). Infine, a partire dalla versione 6.3 di FTD, l'opzione relativa alle dimensioni dei file consente di configurare un file di acquisizione fino a 10 GB. In tal caso, è possibile visualizzare solo il contenuto acquisito in formato pcap.

Per controllare il contenuto dell'acquisizione, è possibile utilizzare un filtro per restringere la ricerca:

```
<#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
```

In questo caso, poiché i pacchetti vengono già tracciati a livello di interfaccia, la causa del rilascio non viene menzionata nell'acquisizione ASP. Tenere presente che un pacchetto può essere tracciato solo in un punto (interfaccia in entrata o rilascio ASP). In tal caso, è consigliabile accettare più rilasci ASP e impostare un motivo di rilascio ASP specifico. Di seguito è riportato un approccio consigliato:

1. Cancella i contatori di rilascio ASP correnti:

```
<#root>
```

```
firepower#
```

```
clear asp drop
```

2. Inviare il flusso che si sta risolvendo attraverso il firewall (eseguire un test).

3. Controllare nuovamente i contatori a discesa ASP e notare che quelli aumentati.

```
<#root>
```

```
firepower#
```

```
show asp drop
```

```
Frame drop:
```

```
No route to host (
```

```
no-route
```

```
)
```

```
234
```

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

```
acl-drop
```

```
)
```

```
71
```

4. Abilitare le acquisizioni ASP per le gocce specifiche rilevate:

```
<#root>
```

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

5. Inviare il flusso per il quale si stanno risolvendo i problemi attraverso il firewall (eseguire un test).

6. Controllare le acquisizioni ASP. In questo caso, i pacchetti sono stati scartati a causa di un percorso assente:

<#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
```

Azione 5. Controllare la tabella delle connessioni Lina FTD.

In alcuni casi, il pacchetto deve uscire dall'interfaccia 'X', ma in altri casi, per qualsiasi motivo, il pacchetto esce dall'interfaccia 'Y'. La determinazione dell'interfaccia di uscita del firewall si basa sul seguente ordine di funzionamento:

1. Ricerca connessione stabilita
2. Ricerca NAT (Network Address Translation) - La fase UN-NAT (NAT di destinazione) ha la precedenza sulla ricerca PBR e route.
3. Policy-Based Routing (PBR)
4. Ricerca nella tabella di routing

Per controllare la tabella di connessione 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

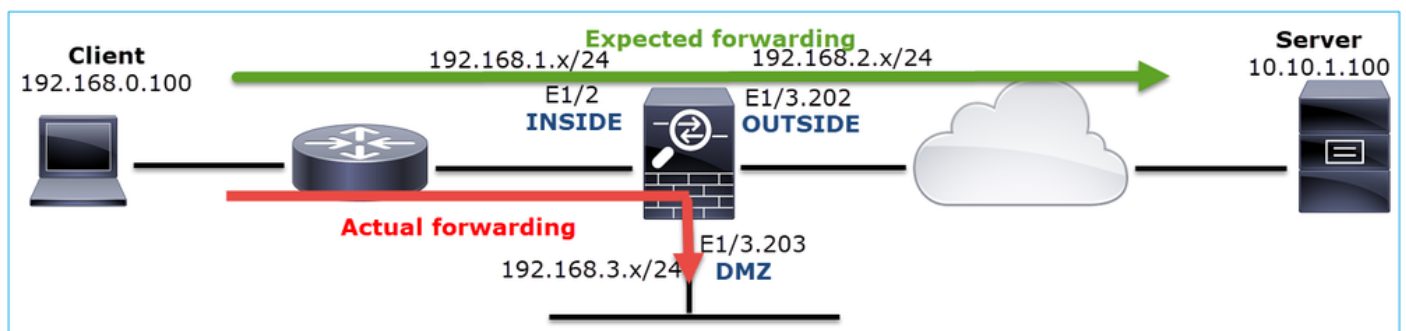
, idle 0:00:01, bytes 0, flags


aA N1


Considerazioni principali:

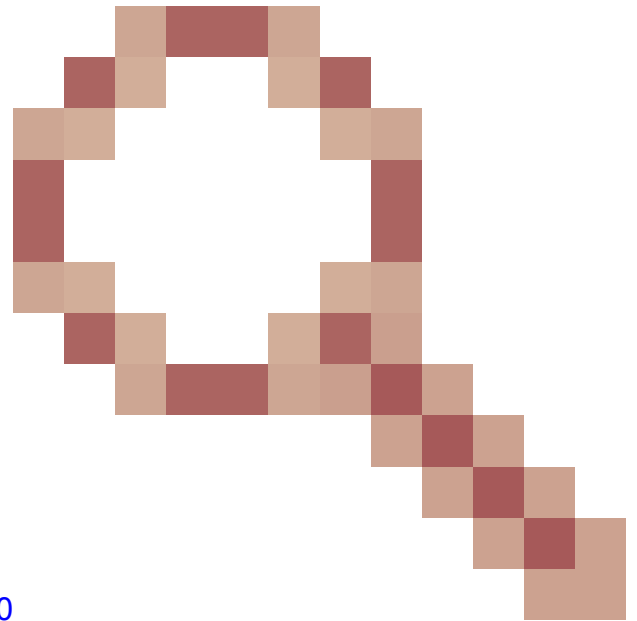
- In base alle bandiere (Aa) la connessione è embrionale (aperta a metà - solo TCP SYN è stato visto dal firewall).
- In base alle porte di origine/destinazione, l'interfaccia in entrata è INSIDE e l'interfaccia in uscita è DMZ.

È possibile visualizzare questa immagine qui:



 Nota: poiché tutte le interfacce FTD hanno un livello di sicurezza 0, l'ordine delle interfacce nell'output show conn si basa sul numero di interfaccia. In particolare, l'interfaccia con il numero vpif-num più alto (numero di interfaccia della piattaforma virtuale) viene selezionata come interna, mentre l'interfaccia con il numero vpif-num più basso viene selezionata come

 esterna. Per visualizzare il valore interface vpif, usare il comando show interface detail.



Miglioramenti correlati, Cisco bug ID [CSCvi15290](#)

ENH: FTD mostra la direzionalità della connessione nell'output 'show conn' 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 partire dal software Firepower versione 6.5, ASA versione 9.13.x, gli output del comando show conn long e show conn detail forniscono informazioni sull'iniziatore e sul risponditore della connessione

Uscita 1

```
<#root>
```

```
firepower#
```

```
show conn long
```

```
...
TCP OUTSIDE: 192.168.2.200/80 (192.168.2.200/80) INSIDE: 192.168.1.100/46050 (192.168.1.100/46050), fla
Initiator: 192.168.1.100, Responder: 192.168.2.200
Connection lookup keyid: 228982375
```

Output 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
```

Inoltre, il comando `show conn long` visualizza gli IP NAT tra parentesi in caso di Network Address Translation:

```
<#root>
firepower#
show conn long
...
TCP OUTSIDE: 192.168.2.222/80 (192.168.2.222/80) INSIDE: 192.168.1.100/34792 (192.168.2.150/34792), fla
Initiator: 192.168.1.100, Responder: 192.168.2.222
Connection lookup keyid: 262895
```

Azione 6. Controllare la cache ARP (Address Resolution Protocol) del firewall.

Se il firewall non è in grado di risolvere l'hop successivo, scarta automaticamente il pacchetto originale (in questo caso TCP SYN) e invia continuamente le richieste ARP finché non risolve l'hop successivo.

Per visualizzare la cache ARP del firewall, utilizzare il comando:

```
<#root>
```

```
firepower#
```

```
show arp
```

Inoltre, per verificare la presenza di host non risolti, è possibile utilizzare il 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
```

Se si desidera controllare ulteriormente l'operazione ARP, è possibile abilitare un'acquisizione specifica per ARP:

```
<#root>
```

```
firepower#
```

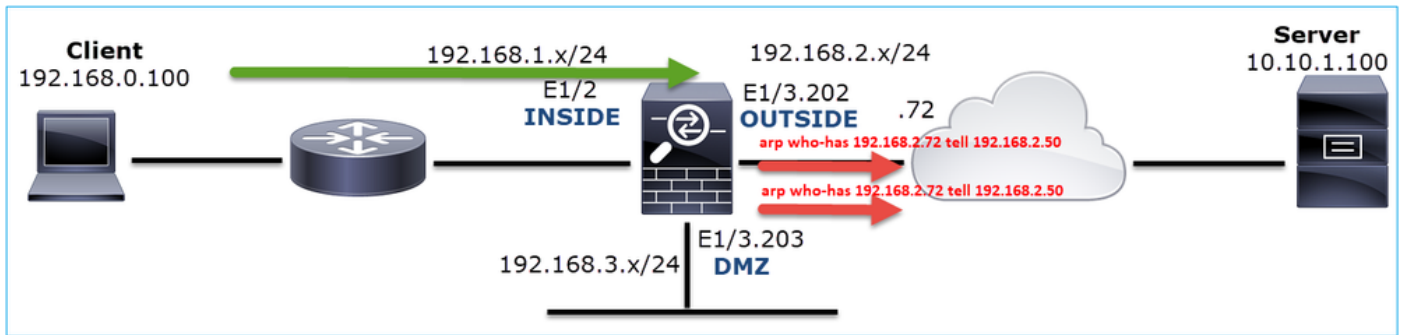
```
capture ARP ethernet-type arp interface OUTSIDE
```

```
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
```

In questo output, il firewall (192.168.2.50) tenta di risolvere l'hop successivo (192.168.2.72), ma non esiste una risposta ARP



L'output riportato di seguito mostra uno scenario funzionale con una risoluzione ARP appropriata:

```
<#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
```

Nel caso in cui non sia presente una voce ARP, una traccia di un pacchetto TCP SYN live mostra:

```
<#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

Come è possibile notare nell'output, il comando trace restituisce Action: allow anche quando l'hop successivo non è raggiungibile e il pacchetto viene scartato automaticamente dal firewall. In questo caso, è necessario controllare anche lo strumento di traccia dei pacchetti perché fornisce un output più accurato:

<#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)

Nelle versioni ASA/Firepower recenti, il messaggio precedente è stato ottimizzato per:

<#root>

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

Check ARP table (show arp) has entry for nexthop

., Drop-location: f

Sintetico delle possibili cause e delle azioni consigliate

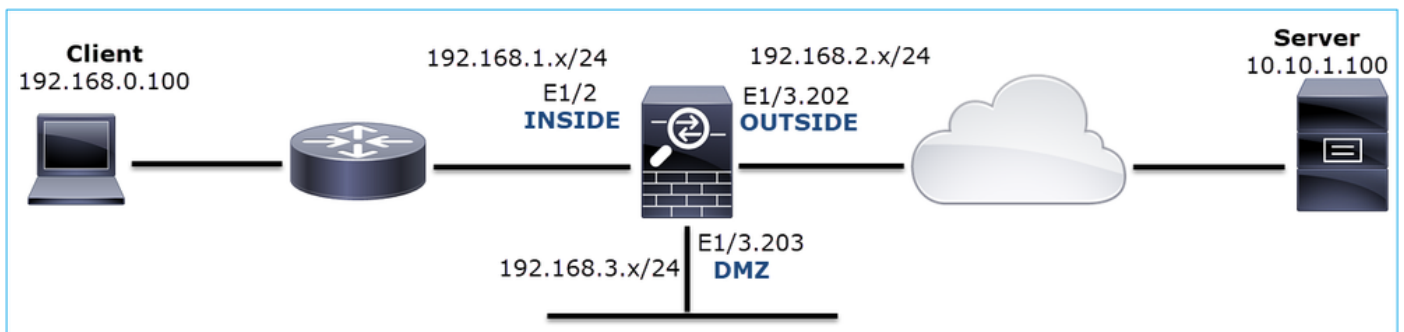
Se si vede solo un pacchetto TCP SYN sulle interfacce in entrata, ma non un pacchetto TCP SYN inviato dall'interfaccia in uscita prevista, alcune possibili cause sono:

Possibile causa	Azioni consigliate
Il pacchetto viene scartato dai criteri di accesso del firewall.	<ul style="list-style-type: none">• Utilizzare packet-tracer o capture w/trace per verificare come il firewall gestisce il pacchetto.• Controllare i registri del firewall.• Controllare le gocce ASP del firewall (show asp drop o tipo di acquisizione asp-drop).• Controllare gli eventi di connessione FMC. In questo caso si presuppone che la registrazione della regola sia abilitata.
Il filtro di acquisizione è errato.	<ul style="list-style-type: none">• Usare packet-tracer o capture w/trace per verificare se esiste una traduzione NAT che modifica l'IP di origine o di destinazione. In tal caso, regolare il filtro di acquisizione.• l'output del comando show conn long visualizza gli IP NAT.
Il pacchetto viene inviato a un'interfaccia di uscita diversa.	<ul style="list-style-type: none">• Usare packet-tracer o capture w/trace per verificare in che modo il firewall gestisce il pacchetto. Tenere presente l'ordine delle operazioni che riguardano la determinazione dell'interfaccia in uscita, la connessione corrente, UN-NAT, PBR e la ricerca nella tabella di routing.• Controllare i registri del firewall.• Controllare la tabella delle connessioni del firewall (show conn). <p>Se il pacchetto viene inviato a un'interfaccia errata perché corrisponde a una connessione corrente, utilizzare il comando clear conn address e specificare la 5-tupla della connessione da cancellare.</p>

Non c'è un percorso verso la destinazione.	<ul style="list-style-type: none"> • Utilizzare packet-tracer o capture w/trace per verificare come il firewall gestisce il pacchetto. • Controllare le cadute ASP del firewall (show asp drop) per il motivo della caduta senza route.
Nessuna voce ARP sull'interfaccia in uscita.	<ul style="list-style-type: none"> • Controllare la cache ARP del firewall (show arp). • Utilizzare packet-tracer per verificare la presenza di un'adiacenza valida.
Interfaccia in uscita inattiva.	Controllare l'output del comando show interface ip brief sul firewall e verificare lo stato dell'interfaccia.

Caso 2. TCP SYN da client, TCP RST da server

Nell'immagine è illustrata la topologia:



Descrizione del problema: HTTP non funziona

Flusso interessato:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocollo: TCP 80

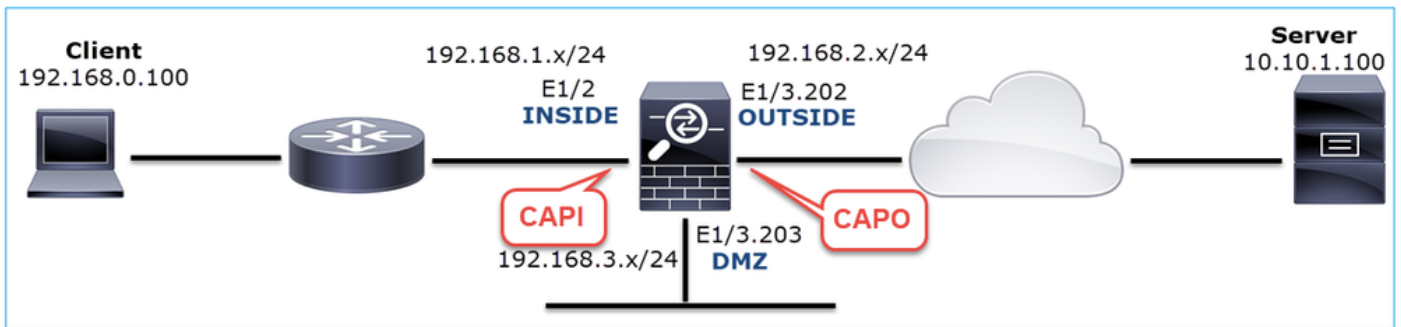
Analisi acquisizione

Abilitare le acquisizioni sul motore LINA 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
```



Acquisizioni - Scenario non funzionale:

L'aspetto delle clip dalla CLI del dispositivo è il seguente:

```
<#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
```

Contenuto 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>
```

...

Contenuto del capo:

<#root>

firepower#

show capture CAPO

```
1: 05:20:36.654507 802.1Q vlan#202 P0 192.168.0.100.22195 > 10.10.1.100.80:
```

S

```
2866789268:2866789268(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>  
2: 05:20:36.904478 802.1Q vlan#202 P0 192.168.0.100.22196 > 10.10.1.100.80:
```

S

```
4785344:4785344(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>  
3: 05:20:36.904997 802.1Q vlan#202 P0 10.10.1.100.80 > 192.168.0.100.22196:
```

R

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

S

```
4235354730:4235354730(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>  
5: 05:20:37.414758 802.1Q vlan#202 P0 10.10.1.100.80 > 192.168.0.100.22196:
```

R

```
0:0(0) ack 4235354731 win 0  
6: 05:20:37.914305 802.1Q vlan#202 P0 192.168.0.100.22196 > 10.10.1.100.80:
```

S

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

Questa immagine mostra l'acquisizione di CAPI in 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	62	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
5	0.000671	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=2476911971 Ack=1 Win=0 Len=0
6	0.499380	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
7	0.000625	10.10.1.100	192.168.0.100	TCP	54	80 → 22196 [RST, ACK] Seq=2853655305 Ack=1 Win=0 Len=0
8	1.739729	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
9	0.000611	10.10.1.100	192.168.0.100	TCP	54	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10	0.499385	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
11	0.000671	10.10.1.100	192.168.0.100	TCP	54	80 → 22195 [RST, ACK] Seq=151733665 Ack=1 Win=0 Len=0

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 > Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
 > Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Seq: 0, Len: 0

Considerazioni principali:

1. L'origine invia un pacchetto TCP SYN.
2. Viene inviato TCP RST verso l'origine.
3. L'origine trasmette nuovamente i pacchetti TCP SYN.
4. Gli indirizzi MAC sono corretti (sui pacchetti in entrata l'indirizzo MAC di origine appartiene al router a valle, l'indirizzo MAC di destinazione appartiene all'interfaccia INSIDE del firewall).

Questa immagine mostra l'acquisizione di CAPO in 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 07:20:36.994478	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)
 > Ethernet II, Src: Cisco_f6:1d:8e (00:be:75:f6:1d:8e), Dst: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8)
 > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
 > Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Seq: 0, Len: 0

Considerazioni principali:

1. L'origine invia un pacchetto TCP SYN.
2. Sull'interfaccia ESTERNA arriva un TCP RST.
3. L'origine trasmette nuovamente i pacchetti TCP SYN.
4. Gli indirizzi MAC sono corretti (sui pacchetti in uscita, il firewall ESTERNO è l'indirizzo MAC di origine, il router a monte è l'indirizzo MAC di destinazione).

Sulla base delle due catture si può concludere che:

- L'handshake a 3 vie TCP tra il client e il server non viene completato
- TCP RST in arrivo sull'interfaccia in uscita del firewall
- Il firewall comunica con i dispositivi upstream e downstream appropriati (in base agli indirizzi MAC)

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Controllare l'indirizzo MAC di origine che invia TCP RST.

Verificare che l'indirizzo MAC di destinazione rilevato nel pacchetto TCP SYN sia lo stesso dell'indirizzo MAC di origine rilevato nel pacchetto TCP RST.

The image displays two screenshots of the Wireshark network protocol analyzer interface, showing a comparison of two network packets. The top screenshot shows a SYN packet (Frame 2) with source MAC address Cisco_f6:1d:8e (00:be:75:f6:1d:8e) and destination MAC address Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8). The bottom screenshot shows a RST, ACK packet (Frame 3) with source MAC address Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8) and destination MAC address Cisco_f6:1d:8e (00:be:75:f6:1d:8e). Arrows indicate the cross-verification of MAC addresses between the two frames.

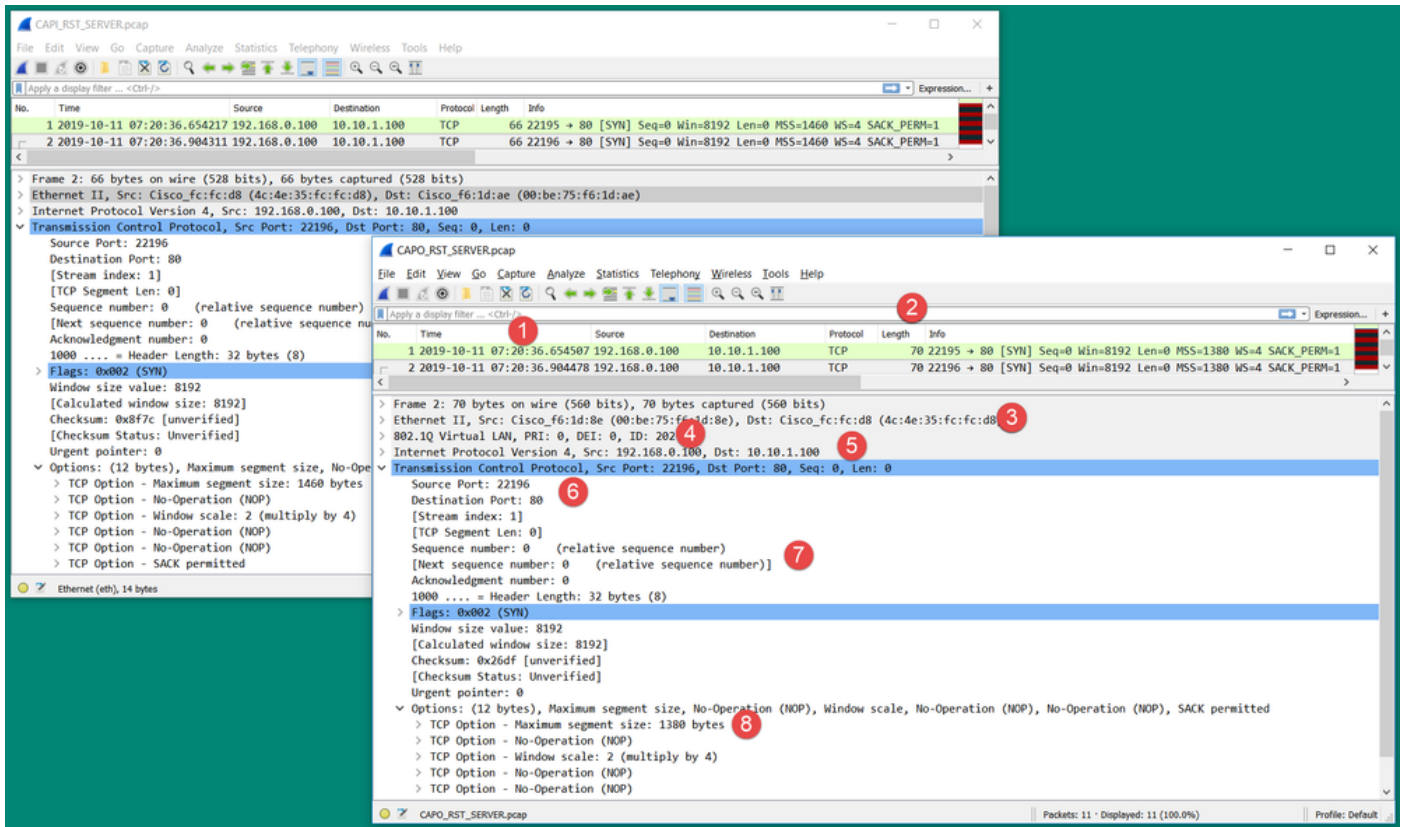
No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-11 07:20:36.654507	192.168.0.100	10.10.1.100	TCP	70	22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
2	2019-10-11 07:20:36.904478	192.168.0.100	10.10.1.100	TCP	70	22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1
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

L'obiettivo di questo controllo è confermare due elementi:

- Verificare che non vi sia alcun flusso asimmetrico.
- Verificare che l'indirizzo MAC appartenga al dispositivo upstream previsto.

Azione 2. Confrontare i pacchetti in entrata e in uscita.

Confrontare visivamente i 2 pacchetti di Wireshark per verificare che il firewall non li modifichi/corrompa. Sono evidenziate alcune differenze previste.



Considerazioni principali:

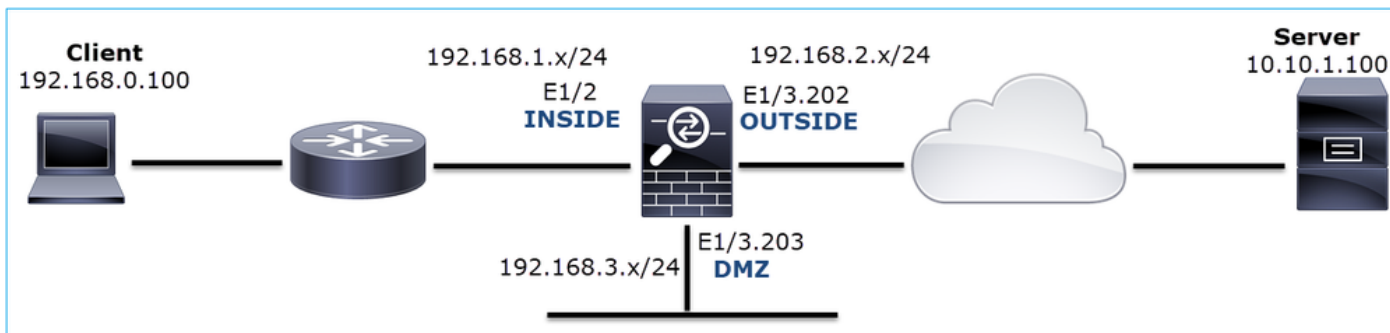
1. I timestamp sono diversi. D'altro canto, la differenza deve essere minima e ragionevole. Ciò dipende dalle funzionalità e dai controlli dei criteri applicati al pacchetto, nonché dal carico sul dispositivo.
2. La lunghezza dei pacchetti varia soprattutto se il firewall aggiunge/rimuove un'intestazione dot1Q.
3. Gli indirizzi MAC sono diversi.
4. L'intestazione dot1Q può essere inserita se l'acquisizione è stata effettuata su una sottointerfaccia.
5. Gli indirizzi IP sono diversi se al pacchetto viene applicato NAT o Port Address Translation (PAT).
6. Le porte di origine o di destinazione sono diverse se al pacchetto viene applicato NAT o PAT.
7. Se si disattiva l'opzione Wireshark Relative Sequence Number, si osserverà che i numeri di sequenza TCP/riconoscimento vengono modificati dal firewall a causa della randomizzazione dell'ISN.
8. Alcune opzioni TCP possono essere sovrascritte. Ad esempio, per impostazione predefinita il firewall modifica il valore TCP Maximum Segment Size (MSS) a 1380 per evitare la frammentazione dei pacchetti nel percorso di transito.

Azione 3. Effettuare una cattura a destinazione.

Se possibile, eseguire una cattura nella stessa destinazione. Se ciò non fosse possibile, eseguire la cattura il più vicino possibile alla destinazione. L'obiettivo è verificare chi invia TCP RST (il server di destinazione è un altro dispositivo nel percorso?).

Caso 3. Handshake TCP a 3 vie + RST da un endpoint

Nell'immagine è illustrata la topologia:



Descrizione del problema: HTTP non funziona

Flusso interessato:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocollo: TCP 80

Analisi acquisizione

Abilitare le acquisizioni sul motore LINA FTD.

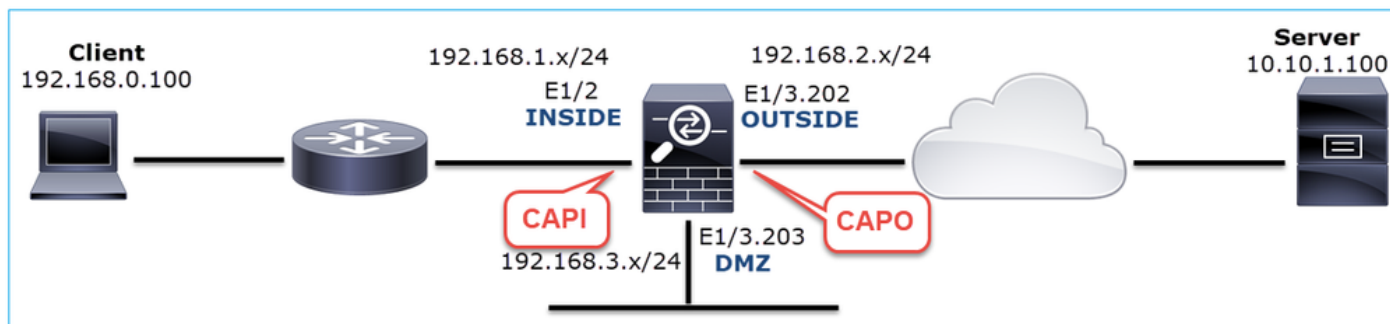
```
<#root>
```

```
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
```



Acquisizioni - Scenario non funzionale:

Ci sono un paio di modi diversi in cui questo problema può manifestarsi nelle clip.

3.1 - Handshake TCP a 3 vie + RST ritardato dal client

Sia il firewall acquisisce CAPI che CAPO contengono gli stessi pacchetti, come mostrato nell'immagine.

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

Considerazioni principali:

1. L'handshake a 3 vie TCP attraversa il firewall.
2. Il server ritrasmette il SYN/ACK.
3. Il client ritrasmette l'ACK.
4. Dopo circa 20 secondi il client si arrende e invia un RST TCP.

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Acquisire le clip il più vicino possibile ai due endpoint.

Le acquisizioni del firewall indicano che l'ACK del client non è stato elaborato dal server. Ciò si basa sui seguenti fatti:

- Il server ritrasmette il SYN/ACK.
- Il client ritrasmette l'ACK.
- Il client invia un TCP RST o FIN/ACK prima di qualsiasi dato.

La cattura sul server mostra il problema. L'ACK client dall'handshake TCP a 3 vie non è mai arrivato:

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 - Handshake TCP a 3 vie + FIN/ACK ritardato dal client + RST ritardato dal server

Sia il firewall acquisisce CAPI che CAPO contengono gli stessi pacchetti, come mostrato nell'immagine.

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

Considerazioni principali:

1. L'handshake a 3 vie TCP attraversa il firewall.
2. Dopo circa 5 sec il client invia un messaggio FIN/ACK.
3. Dopo circa 20 secondi, il server si arrende e invia un RST TCP.

In base a questa acquisizione si può concludere che, anche se esiste un handshake a 3 vie TCP attraverso il firewall, sembra che non venga mai effettivamente completato su un endpoint (le ritrasmissioni indicano questa condizione).

Azioni consigliate

Come nel caso 3.1

3.3 - Handshake TCP a 3 vie + RST ritardato dal client

Sia il firewall acquisisce CAPI che CAPO contengono gli stessi pacchetti, come mostrato nell'immagine.

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

Considerazioni principali:

1. L'handshake a 3 vie TCP attraversa il firewall.
2. Dopo circa 20 secondi il client si arrende e invia un RST TCP.

Sulla base di tali acquisizioni si può concludere che:

- Dopo 5-20 secondi, un endpoint rinuncia e decide di terminare la connessione.

Azioni consigliate

Come nel caso 3.1

3.4 - Handshake TCP a 3 vie + RST immediato dal server

Entrambi i firewall acquisiscono CAPI e CAPO contengono questi pacchetti, come mostrato nell'immagine.

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

Considerazioni principali:

1. L'handshake a 3 vie TCP attraversa il firewall.
2. Il server invia un segnale TCP RST pochi millisecondi dopo il pacchetto ACK.

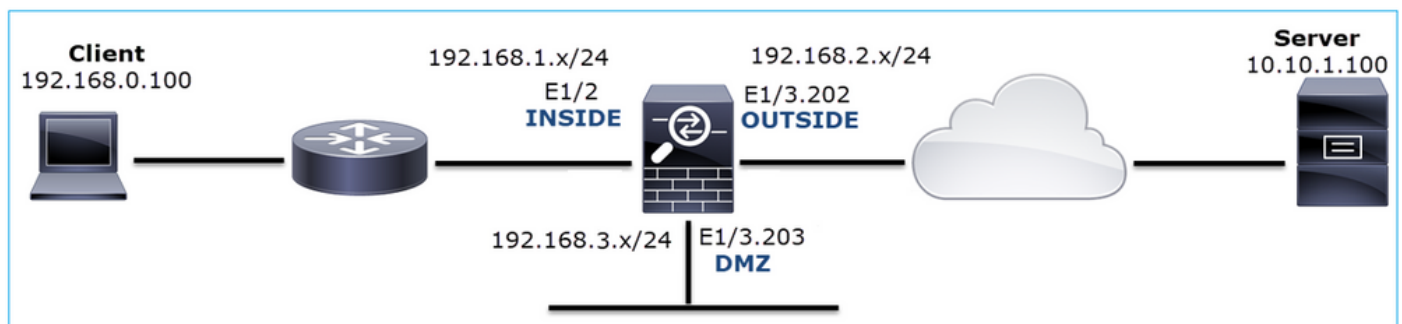
Azioni consigliate

Azione: acquisire le clip il più vicino possibile al server.

Un RST TCP immediato dal server potrebbe indicare un server malfunzionante o un dispositivo nel percorso che invia l'RST TCP. Eseguire un'acquisizione sul server stesso e determinare l'origine di TCP RST.

Caso 4. TCP RST dal client

Nell'immagine è illustrata la topologia:



Descrizione del problema: HTTP non funziona.

Flusso interessato:

Src IP: 192.168.0.100

Dst IP: 10.10.1.100

Protocollo: TCP 80

Analisi acquisizione

Abilita le acquisizioni sul motore LINA FTD.

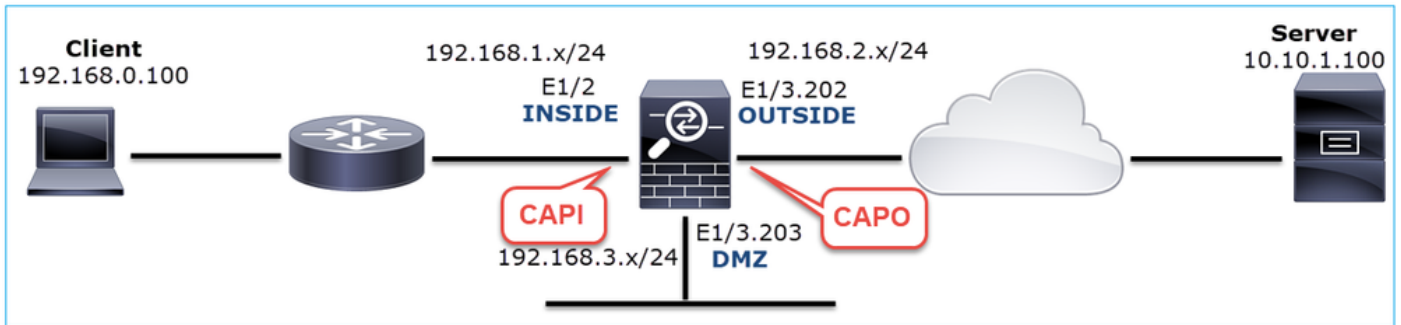
```
<#root>
```

```
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
```



Acquisizioni - Scenario non funzionale:

Questi sono i contenuti 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

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

```

I registri del firewall visualizzano:

```
<#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
```

```

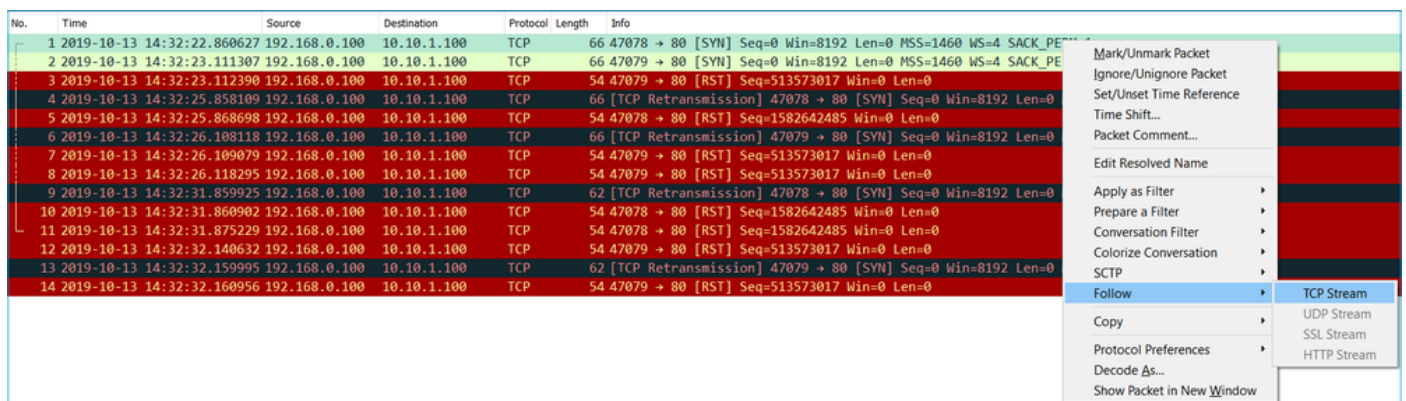
Oct 13 2019 13:57:45: %FTD-6-302013: Built inbound TCP connection 4871 for INSIDE:192.168.0.100/47741 (
Oct 13 2019 13:57:45: %FTD-6-302014: Teardown TCP connection 4871 for INSIDE:192.168.0.100/47741 to OUT

```

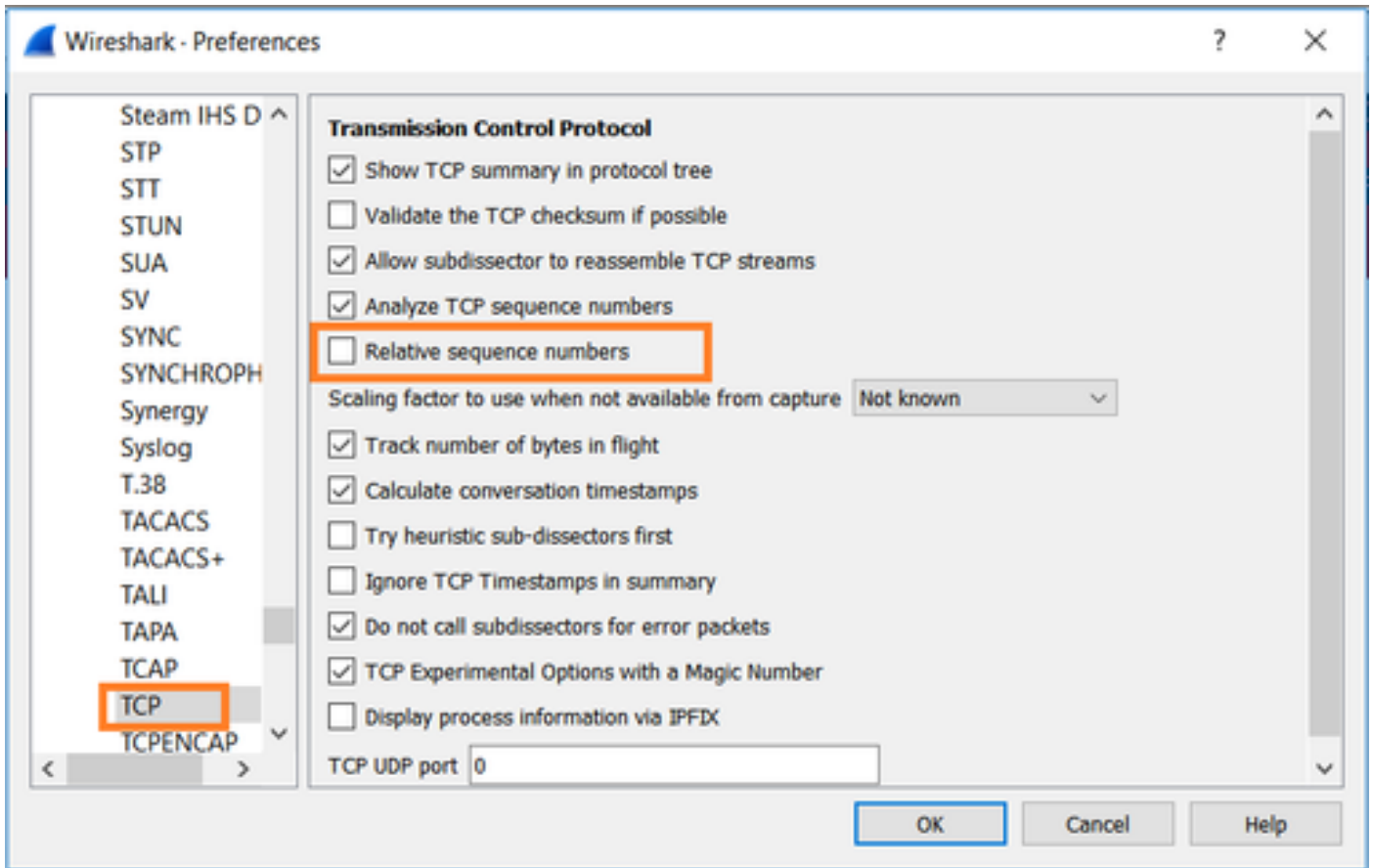
Questi registri indicano la presenza di una RST TCP che arriva all'interfaccia INSIDE del firewall

CAPI di Wireshark:

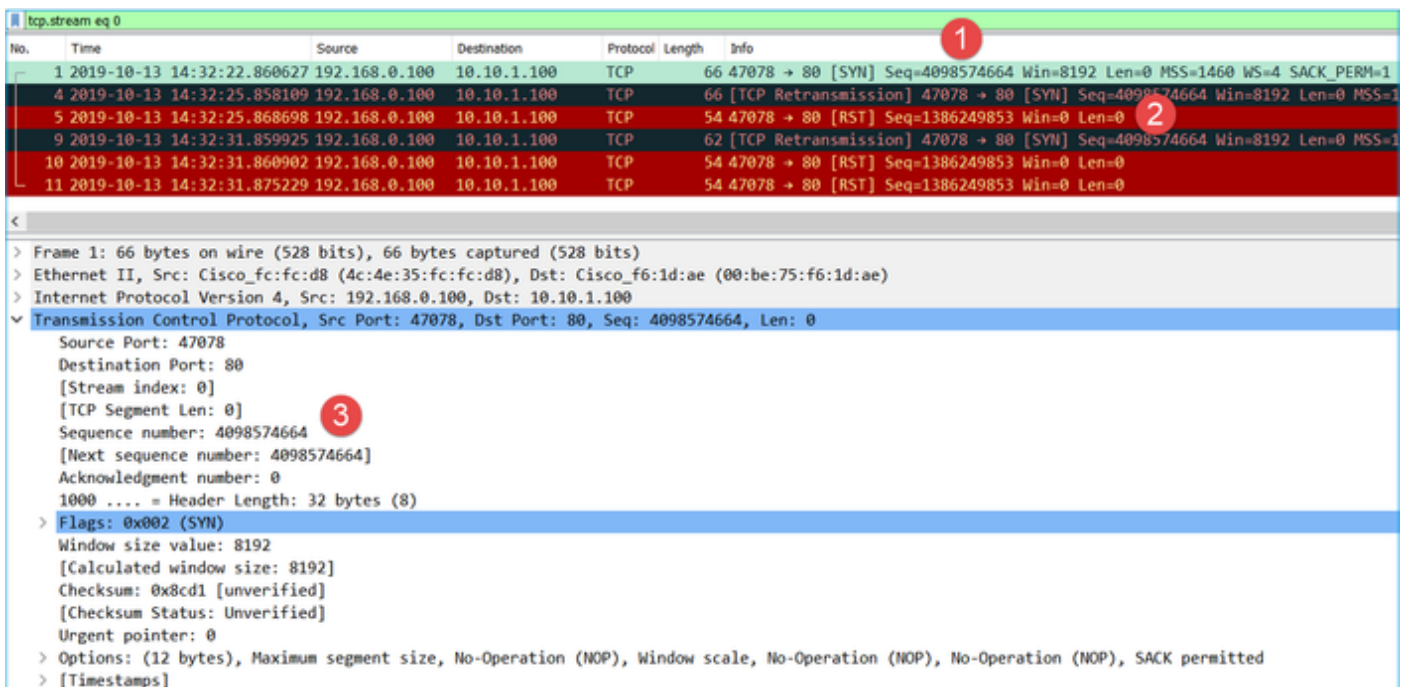
Seguire il primo flusso TCP, come mostrato nell'immagine.



In Wireshark, selezionare Modifica > Preferenze > Protocolli > TCP e deselezionare l'opzione Numeri di sequenza relativi come mostrato nell'immagine.



Questa immagine mostra il contenuto del primo flusso in CAPI capture:



Considerazioni principali:

1. Il client invia un pacchetto TCP SYN.
2. Il client invia un pacchetto TCP RST.
3. Il valore del numero di sequenza del pacchetto TCP SYN è uguale a 4098574664.

Lo stesso flusso nell'acquisizione 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

Considerazioni principali:

1. Il client invia un pacchetto TCP SYN. Il firewall rende casuale l'ISDN.
2. Il client invia un pacchetto TCP RST.

Sulla base delle due acquisizioni si può concludere che:

- Non esiste alcun handshake TCP a 3 vie tra il client e il server.
- È presente un RST TCP che proviene dal client. Il valore del numero di sequenza TCP RST nell'acquisizione CAPI è 1386249853.

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Acquisisci un'immagine sul client.

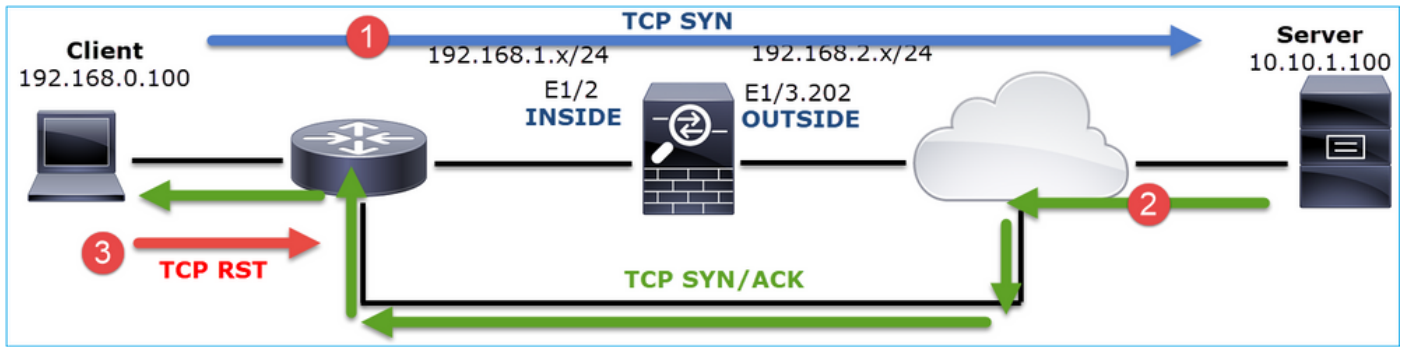
In base alle acquisizioni raccolte sul firewall, esiste una forte indicazione di un flusso asimmetrico. Ciò si basa sul fatto che il client invia un RST TCP con un valore di 1386249853 (l'ISN randomizzato):

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

Considerazioni principali:

1. Il client invia un pacchetto TCP SYN. Il numero di sequenza è 4098574664 ed è lo stesso di quello visualizzato sull'interfaccia CAPI (Firewall INSIDE Interface)
2. È presente un TCP SYN/ACK con numero ACK 1386249853 (previsto a causa della randomizzazione ISDN). Il pacchetto non è stato rilevato nelle acquisizioni del firewall
3. Il client invia una richiesta RST TCP perché si aspettava un valore SYN/ACK con numero ACK pari a 4098574665, ma ha ricevuto il valore di 1386249853

Ciò può essere visualizzato come:

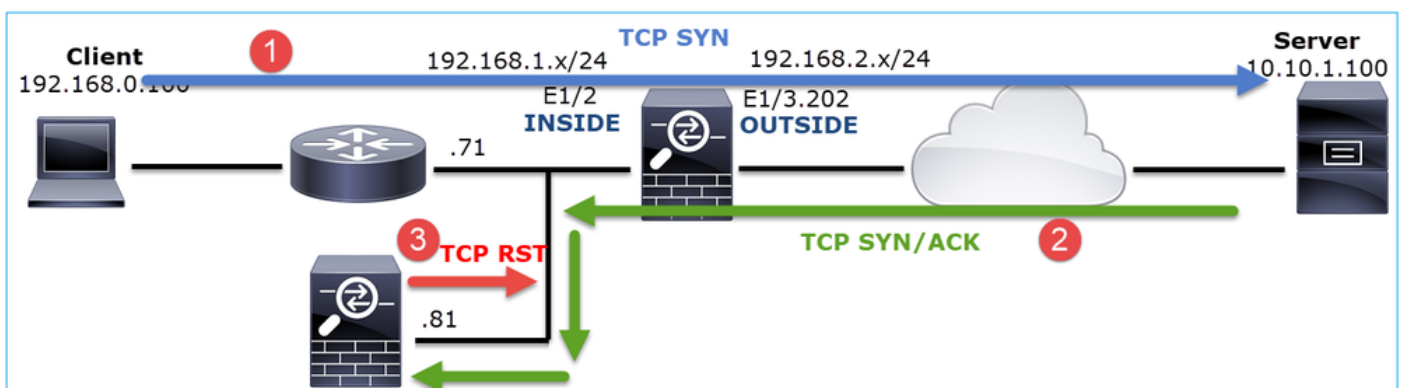


Azione 2. Controllare il routing tra il client e il firewall.

Confermare che:

- Gli indirizzi MAC visualizzati nelle clip sono quelli previsti.
- Verificare che il routing tra il firewall e il client sia simmetrico.

In alcuni scenari, l'RST proviene da un dispositivo situato tra il firewall e il client mentre nella rete interna è presente un routing asimmetrico. Nell'immagine viene in genere visualizzato un caso:



In questo caso, l'acquisizione ha questo contenuto. Si noti la differenza tra l'indirizzo MAC di origine del pacchetto TCP SYN e l'indirizzo MAC di origine del pacchetto TCP RST e l'indirizzo MAC di destinazione del pacchetto 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. Trasferimento TCP lento (scenario 1)

Descrizione del problema:

Il trasferimento SFTP tra gli host 10.11.4.171 e 10.77.19.11 è lento. Sebbene la larghezza di banda minima (BW) tra i 2 host sia di 100 Mbps, la velocità di trasferimento non supera i 5 Mbps.

Allo stesso tempo, la velocità di trasferimento tra gli host 10.11.2.124 e 172.25.18.134 è notevolmente superiore.

Nozioni di base:

La velocità massima di trasferimento per una singola connessione TCP è determinata dal BDP (Bandwidth Delay Product). La formula utilizzata è illustrata nell'immagine:

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

Per ulteriori informazioni sul BDP, consultare le risorse disponibili al seguente indirizzo:

- [Perché l'applicazione utilizza solo 10 Mbps Anche il collegamento è 1 Gbps?](#)
- [BRKSEC-3021 - Avanzate - Ottimizzazione delle prestazioni del firewall](#)

Scenario 1. Trasferimento lento

Nell'immagine è illustrata la topologia:



Flusso interessato:

Src IP: 10.11.4.171

Dst IP: 10.7.19.11

Protocollo: SFTP (FTP over SSH)

Analisi acquisizione

Abilita acquisizioni sul motore LINA 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
```

 Avviso: le acquisizioni LINA sulle acquisizioni FP1xxx e FP21xx influiscono sulla velocità di trasferimento del traffico che attraversa l'FTD. Non abilitare le acquisizioni LINA sulle piattaforme FP1xxx e FP21xxx quando si risolvono problemi di prestazioni (trasferimento lento tramite FTD). Usare invece SPAN o un dispositivo HW Tap in aggiunta alle acquisizioni sugli host di origine e di destinazione. Il problema è documentato nell'ID bug Cisco [CSCvo30697](https://www.cisco.com/c/en-us/bugtools/bugtools/bugtools.html?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.
```

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Calcolo del tempo di andata e ritorno (RTT)

Identificare innanzitutto il flusso di trasferimento e seguirlo:

No.	Time	Source	Destination	Protocol	Length	Window size value
1	0.000000	10.11.4.171	10.77.19.11	TCP	70	49640
2	0.072521	10.77.19.11	10.11.4.171	TCP	70	49680
3	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680
4	0.077068	10.77.19.11	10.11.4.171	SSHv2	80	49680
5	0.000152	10.11.4.171	10.77.19.11	TCP	58	49680
6	0.000244	10.11.4.171	10.77.19.11	SSHv2	80	49680
7	0.071545	10.77.19.11	10.11.4.171	TCP	58	49680
8	0.000153	10.11.4.171	10.77.19.11	SSHv2	538	49680
9	0.041288	10.77.19.11	10.11.4.171	SSHv2	738	49680
10	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680
11	0.030165	10.77.19.11	10.11.4.171	TCP	58	49680
12	0.000168	10.11.4.171	10.77.19.11	SSHv2	82	49680

View	Info
Follow	TCP Stream
Copy	UDP Stream
Protocol Preferences	SSL Stream
	HTTP Stream

Cambiate la vista di Wireshark in modo da visualizzare i secondi trascorsi dal precedente pacchetto visualizzato. Questo semplifica il calcolo dell'RTT:

View	Info
Full Screen	Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+1
Packet List	Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
Packet Details	Time of Day (01:02:03.123456) Ctrl+Alt+2
Packet Bytes	Seconds Since 1970-01-01 Ctrl+Alt+3
Time Display Format	Seconds Since Beginning of Capture Ctrl+Alt+4
Name Resolution	Seconds Since Previous Captured Packet Ctrl+Alt+5
Zoom	Seconds Since Previous Displayed Packet Ctrl+Alt+6
Expand Subtrees	
Collapse Subtrees	
Expand All	

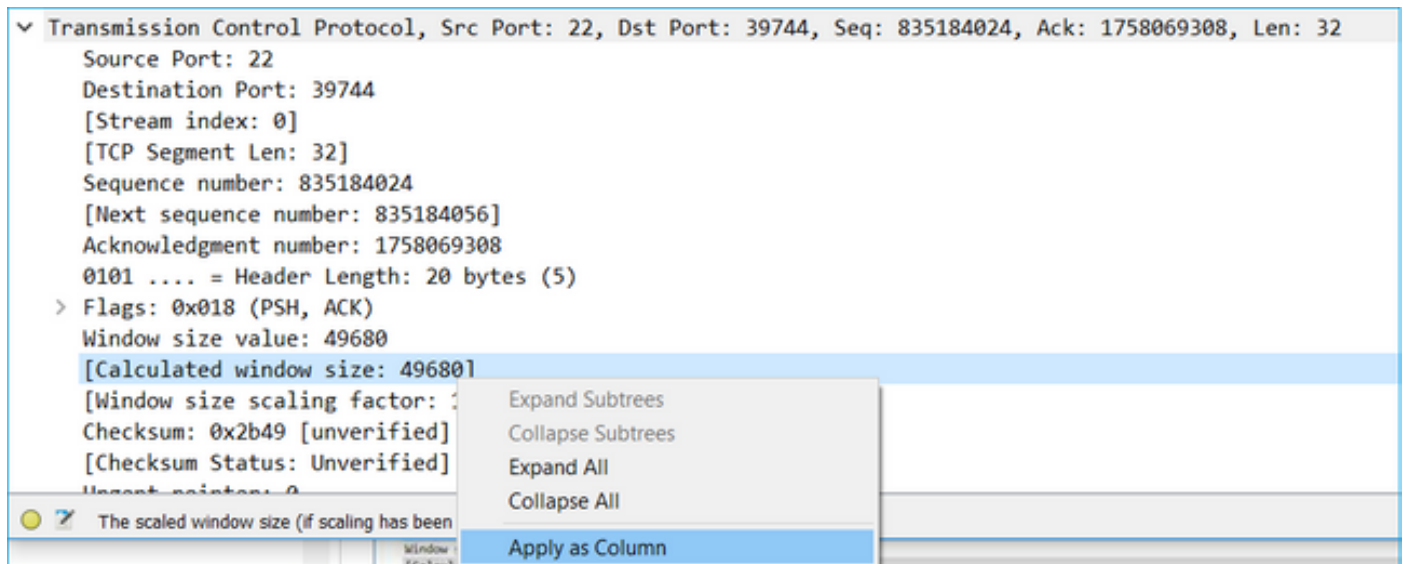
L'RTT può essere calcolato sommando i valori temporali tra due scambi di pacchetti (uno verso l'origine e uno verso la destinazione). In questo caso, il pacchetto 2 mostra il segnale RTT tra il firewall e il dispositivo che ha inviato il pacchetto SYN/ACK (server). Nel pacchetto 3 viene mostrato il segnale RTT tra il firewall e il dispositivo che ha inviato il pacchetto ACK (client). L'aggiunta dei due numeri fornisce una buona stima dell'RTT end-to-end:

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 msec

Calcolo dimensioni finestra TCP

Espandere un pacchetto TCP, espandere l'intestazione TCP, selezionare Dimensione calcolata finestra e selezionare Applica come colonna:



Controllare la colonna Valore dimensioni finestra calcolate per verificare il valore massimo delle dimensioni della finestra durante la sessione TCP. È inoltre possibile selezionare il nome della colonna e ordinare i valori.

Se si verifica il download di un file (server > client), è necessario controllare i valori annunciati dal server. Il valore massimo delle dimensioni della finestra annunciato dal server determina la velocità di trasferimento massima raggiunta.

In questo caso, le dimensioni della finestra TCP sono ≈ 50000 byte

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)

In base a questi valori e con l'uso della formula Prodotto ritardo larghezza di banda si ottiene la massima larghezza di banda teorica che può essere raggiunta in queste condizioni: $50000 * 8 / 0.08 = 5 \text{ Mbps}$ larghezza di banda teorica massima.

In questo caso, corrisponde a ciò che il client sperimenta.

Controllare attentamente l'handshake TCP a 3 vie. Entrambi i lati, e soprattutto il server, annunciano un valore di scala della finestra pari a 0 che significa $2^0 = 1$ (nessuna scala delle

finestre). Questo influisce negativamente sulla velocità di trasferimento:

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) on interface 0
> 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)
```

A questo punto, è necessario eseguire un'acquisizione sul server, verificare che sia quello che annuncia la scala della finestra = 0 e riconfigurarla (consultare la documentazione del server per informazioni su come eseguire questa operazione).

Scenario 2. Trasferimento rapido

Esaminiamo ora lo scenario positivo (trasferimento rapido attraverso la stessa rete):

Topologia:



Il flusso di interessi:

Src IP: 10.11.2.124

Dst IP: 172.25.18.134

Protocollo: SFTP (FTP over SSH)

Abilita acquisizioni sul motore LINA 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

Calcolo del tempo di andata e ritorno (RTT, Round Trip Time): In questo caso, il valore RTT è \approx 300 msec.

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

Calcolo dimensioni finestra TCP: il server annuncia un fattore di scala della finestra TCP pari a 7.

```
> Internet Protocol Version 4, Src: 172.25.18.134, Dst: 10.11.2.124
v Transmission Control Protocol, Src Port: 22, Dst Port: 57093, Seq: 661963571, Ack: 1770516295, Len: 0
  Source Port: 22
  Destination Port: 57093
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 661963571
  [Next sequence number: 661963571]
  Acknowledgment number: 1770516295
  1010 ... = Header Length: 40 bytes (10)
  > Flags: 0x012 (SYN, ACK)
  Window size value: 14480
  [Calculated window size: 14480]
  Checksum: 0x6497 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  v Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
    > TCP Option - Maximum segment size: 1300 bytes
    > TCP Option - SACK permitted
    > TCP Option - Timestamps: TSval 390233290, TSecr 981659424
    > TCP Option - No-Operation (NOP)
    > TCP Option - Window scale: 7 (multiply by 128)
  > [SEQ/ACK analysis]
```

Le dimensioni della finestra TCP del server sono \approx 1600000 byte:

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=

In base a questi valori, la formula Ritardo larghezza di banda prodotto fornisce:

$1600000 * 8 / 0,3 =$ velocità teorica massima di trasferimento 43 Mbps

Caso 6. Trasferimento TCP lento (scenario 2)

Descrizione del problema: il trasferimento di file FTP (download) attraverso il firewall è lento.

Nell'immagine è illustrata la topologia:



Flusso interessato:

Src IP: 192.168.2.220

Dst IP: 192.168.1.220

Protocollo: FTP

Analisi acquisizione

Abilitare le acquisizioni sul motore LINA 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
```

Selezionare un pacchetto FTP-DATA e seguire il canale dati FTP sull'acquisizione CAPI (FTD INSIDE capture):

75	0.000412	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670018383
76	0.000518	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
77	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
78	0.000046	192.168.1.220	192.168.2.220	FTP-DATA	not captured] FTP Data: 124
79	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
80	0.000107	192.168.2.220	192.168.1.220	TCP	Seq=1884231612 Ack=2670019631
81	0.000092	192.168.2.220	192.168.1.220	TCP	Seq=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	38 → 54494 [ACK] Seq=2670026
89	0.000397	192.168.2.220	192.168.1.220	TCP	Seq=2670027119
90	0.000869	192.168.1.220	192.168.2.220	FTP-DATA	(RETR file15mb)

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

Contenuto del flusso 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=3577289500 TSecr=0 WS=128
28	1.026564	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
29	1.981594	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2669999678 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=2669999679 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
34	0.001617	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
35	0.000351	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669999927 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=2669999927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699993423
39	0.000077	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669999927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=26699994671
40	0.309906	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669999927 Ack=1884231612 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
41	0.000488	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=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	0.000300	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	192.168.2.220	192.168.1.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
54	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
55	0.000199	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670002159 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
56	0.000229	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
57	0.000183	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58	0.000106	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=65280 Len=0 TSval=3577292742 TSecr=4264507 SLE=26700004655 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=26700004655 SRE=26700008399
60	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

Il contenuto dell'acquisizione di CAPO:

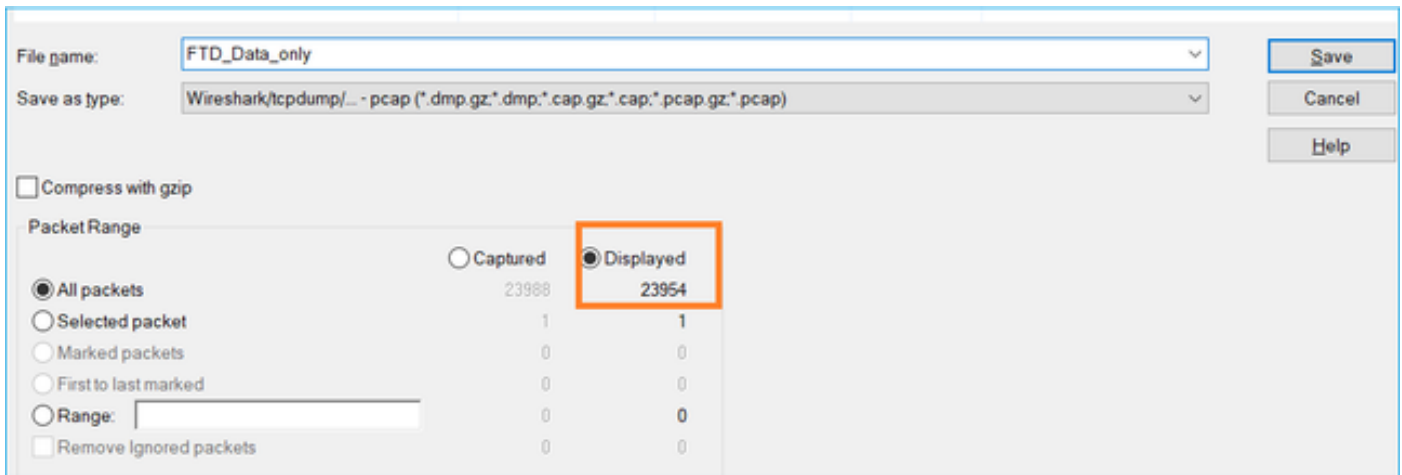
31	0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
33	1.026534	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
34	1.981400	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
35	0.000610	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
38	0.001328	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
40	0.000641	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 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=2224321804 SRE=2224331904
45	0.309905	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	0.000590	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	192.168.2.220	192.168.1.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
49	0.000076	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
50	0.000290	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
51	0.000046	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=2224324400 SRE=2224326896
52	0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000351	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2224324400 SRE=2224328144
54	0.918019	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224323152 Ack=2157030682 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
55	0.001007	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224328144 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
56	0.000457	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
57	0.000061	192.168.2.220	192.168.1.220	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)

Considerazioni principali:

1. Sono presenti pacchetti TCP non in ordine (OO).
2. Ritrasmissione TCP.
3. C'è un'indicazione di una perdita di pacchetto (pacchetti scartati).

 Suggerimento: salvare le clip mentre si passa a File > Esporta pacchetti specificati. Quindi

salva solo l'intervallo di pacchetti visualizzati



Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Identificare l'ubicazione di perdita del pacchetto.

In casi come questo, è necessario acquisire immagini simultaneamente e usare la metodologia "divide and conquista" per identificare i segmenti di rete che causano la perdita di pacchetti. Dal punto di vista del firewall, ci sono 3 scenari principali:

1. La perdita di pacchetti è causata dal firewall stesso.
2. La perdita di pacchetti è causata a valle del dispositivo firewall (direzione da server a client).
3. La perdita di pacchetti è causata a monte del dispositivo firewall (direzione dal client al server).

Perdita di pacchetti causata dal firewall: per stabilire se la perdita di pacchetti è causata dal firewall, è necessario confrontare l'acquisizione in entrata con l'acquisizione in uscita. Ci sono molti modi per confrontare 2 diverse clip. In questa sezione viene illustrato un modo per eseguire questa operazione.

Procedura per confrontare 2 acquisizioni e identificare la perdita di pacchetti

Passaggio 1. Verificare che le 2 clip contengano pacchetti provenienti dalla stessa finestra temporale. Ciò significa che in un'acquisizione non devono essere presenti pacchetti acquisiti prima o dopo l'altra. A tale scopo, è possibile procedere in diversi modi:

- Controllare il primo e l'ultimo valore di identificazione (ID) dell'IP del pacchetto.
- Controllare i valori del primo e dell'ultimo timestamp del pacchetto.

In questo esempio si nota che i primi pacchetti di ciascuna acquisizione hanno gli stessi valori di ID 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	0xb034 (2612)	54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
2	2019-10-16 16:13:45.195958	192.168.2.220	192.168.1.220	TCP	74	0xb035 (2613)	[TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288526 TSecr=0 WS=128
3	2019-10-16 16:13:47.177542	192.168.1.220	192.168.2.220	TCP	66	0xb036 (2614)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
4	2019-10-16 16:13:47.178030	192.168.2.220	192.168.1.220	TCP	66	0xb036 (2614)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
5	2019-10-16 16:13:47.179647	192.168.1.220	192.168.2.220	TCP	1314	0x1521 (5409)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
6	2019-10-16 16:13:47.179998	192.168.2.220	192.168.1.220	TCP	66	0xb037 (2615)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
7	2019-10-16 16:13:47.180456	192.168.1.220	192.168.2.220	TCP	1314	0x1522 (5411)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
8	2019-10-16 16:13:47.180517	192.168.1.220	192.168.2.220	TCP	1314	0x1524 (5412)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
9	2019-10-16 16:13:47.180715	192.168.2.220	192.168.1.220	TCP	78	0xb038 (2616)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
10	2019-10-16 16:13:47.180792	192.168.2.220	192.168.1.220	TCP	78	0xb039 (2617)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
11	2019-10-16 16:13:47.489888	192.168.1.220	192.168.2.220	TCP	1314	0x1525 (5413)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
12	2019-10-16 16:13:47.490376	192.168.2.220	192.168.1.220	TCP	66	0xb03a (2618)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
13	2019-10-16 16:13:47.490865	192.168.1.220	192.168.2.220	TCP	1314	0x1526 (5414)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
14	2019-10-16 16:13:47.490910	192.168.2.220	192.168.1.220	TCP	74	0x151f (5407)	2388 → 54494 [SYN, ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
15	2019-10-16 16:13:47.490987	192.168.1.220	192.168.2.220	TCP	1314	0x1529 (5417)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
16	2019-10-16 16:13:47.491231	192.168.2.220	192.168.1.220	TCP	66	0xb03b (2619)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
17	2019-10-16 16:13:47.491261	192.168.2.220	192.168.1.220	TCP	78	0xb03c (2620)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
18	2019-10-16 16:13:47.491765	192.168.1.220	192.168.2.220	TCP	1314	0x152a (5418)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
19	2019-10-16 16:13:47.492024	192.168.2.220	192.168.1.220	TCP	78	0xb03d (2621)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
20	2019-10-16 16:13:48.410150	192.168.1.220	192.168.2.220	TCP	1314	0x152e (5422)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
21	2019-10-16 16:13:48.411050	192.168.2.220	192.168.1.220	TCP	66	0xb03e (2622)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
22	2019-10-16 16:13:48.411569	192.168.1.220	192.168.2.220	TCP	1314	0x152f (5423)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
23	2019-10-16 16:13:48.411630	192.168.2.220	192.168.1.220	TCP	1314	0x1530 (5424)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
24	2019-10-16 16:13:48.411654	192.168.1.220	192.168.2.220	TCP	1314	0x1532 (5426)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
25	2019-10-16 16:13:48.411660	192.168.2.220	192.168.1.220	TCP	1314	0x1533 (5427)	2388 → 54494 [ACK] Seq=2669988678 Ack=1884231612 Win=8192 Len=0 MSS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
26	2019-10-16 16:13:48.411859	192.168.2.220	192.168.1.220	TCP	66	0xb03f (2623)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384
27	2019-10-16 16:13:48.412088	192.168.2.220	192.168.1.220	TCP	66	0xb040 (2624)	54494 → 2388 [ACK] Seq=1884231612 Ack=2669988678 Win=29312 Len=0 TSval=3572291568 TSecr=4264384

In caso contrario:

1. Confrontare gli indicatori orari del primo pacchetto di ciascuna acquisizione.
2. Dall'acquisizione con l'ultimo Timestamp ottenere un filtro da esso modificare il filtro Timestamp da == a >= (il primo pacchetto) e <= (l'ultimo pacchetto), ad esempio:

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

Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

Encapsulation type: Ethernet (1)

Arrival Time: Oct 16, 2019 16:13:43.245638000 Central European Daylight Time

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1571235223.245638000 seconds

[Time delta from previous captured frame: 0.000000000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 0.000000000 seconds]

Frame Number: 2

Frame Length: 74 bytes (592 bits)

Capture Length: 74 bytes (592 bits)

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

Selected

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

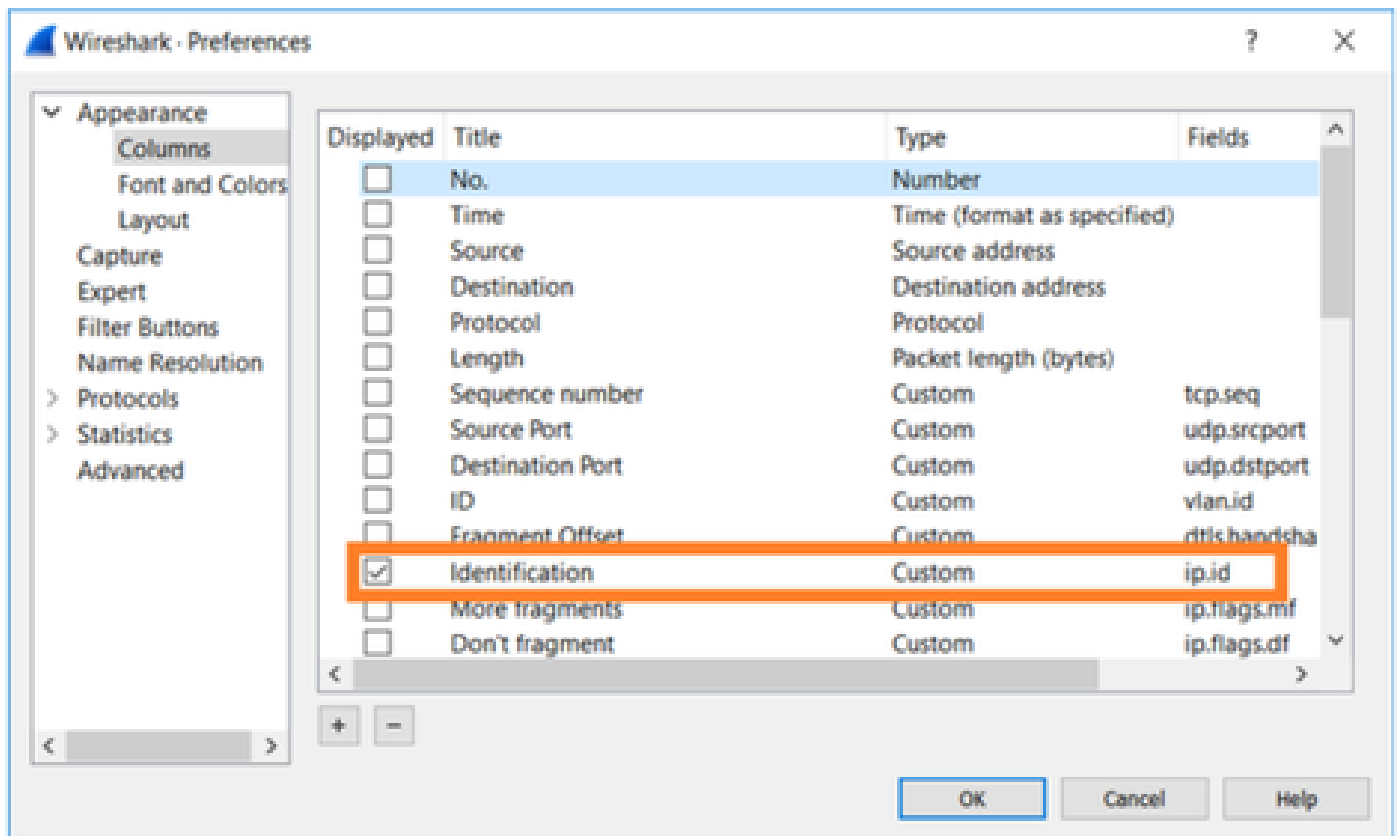
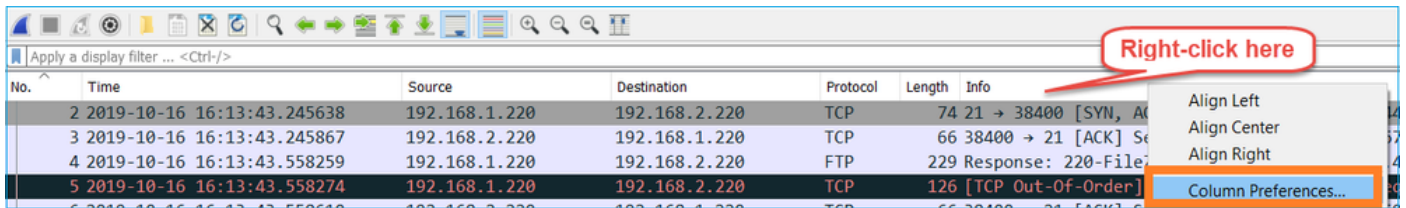
3. Esportare i pacchetti specificati in una nuova acquisizione, selezionare File > Esporta pacchetti specificati, quindi salvare i pacchetti visualizzati. A questo punto, entrambe le clip devono contenere pacchetti che coprono la stessa finestra temporale. A questo punto è possibile iniziare il confronto delle due clip.

Passaggio 2. Specificare il campo del pacchetto da utilizzare per il confronto tra le due acquisizioni. Esempio di campi utilizzabili:

- Identificazione IP
- Numero di sequenza RTP
- Numero di sequenza ICMP

Creare una versione di testo di ciascuna acquisizione contenente il campo per ciascun pacchetto

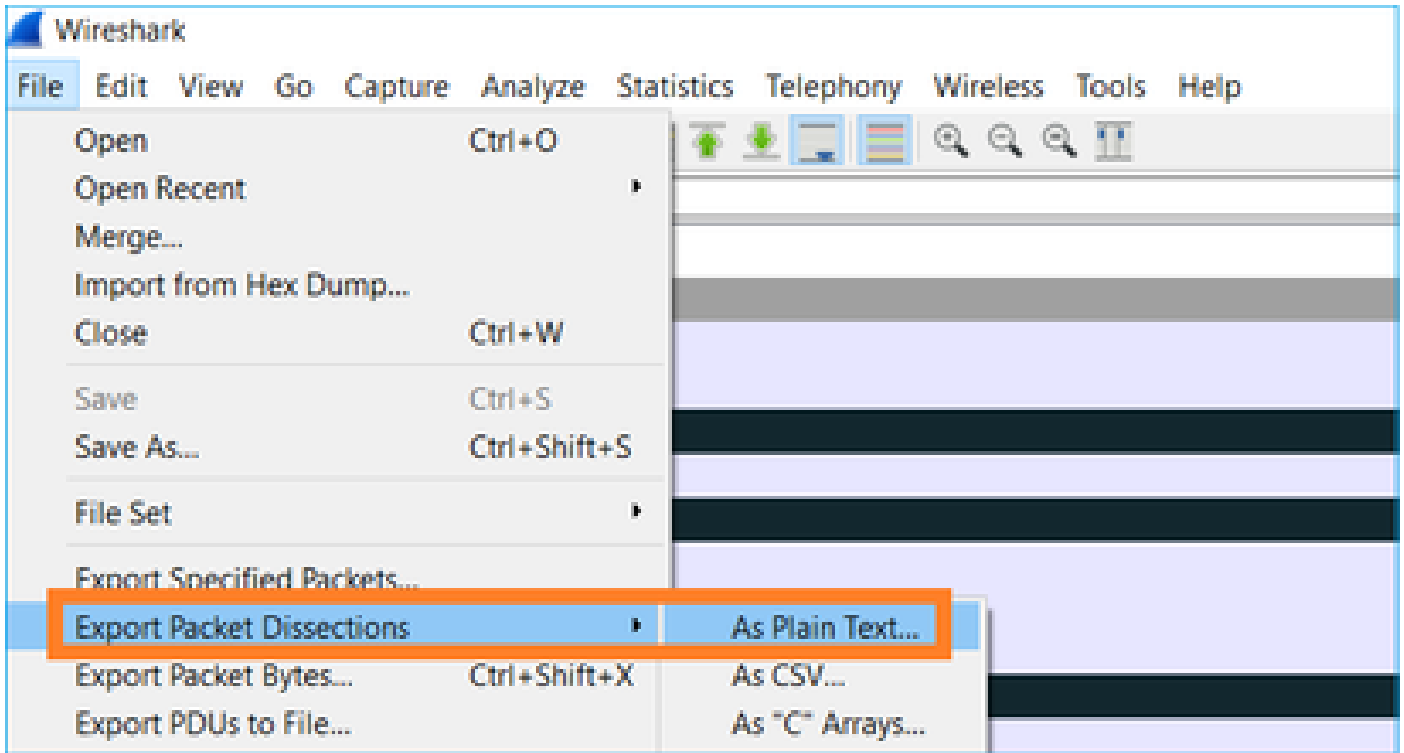
specificato nel passaggio 1. A tale scopo, lasciare solo la colonna di interesse; ad esempio, se si desidera confrontare i pacchetti basati sull'identificazione IP, modificare l'acquisizione come mostrato nell'immagine.



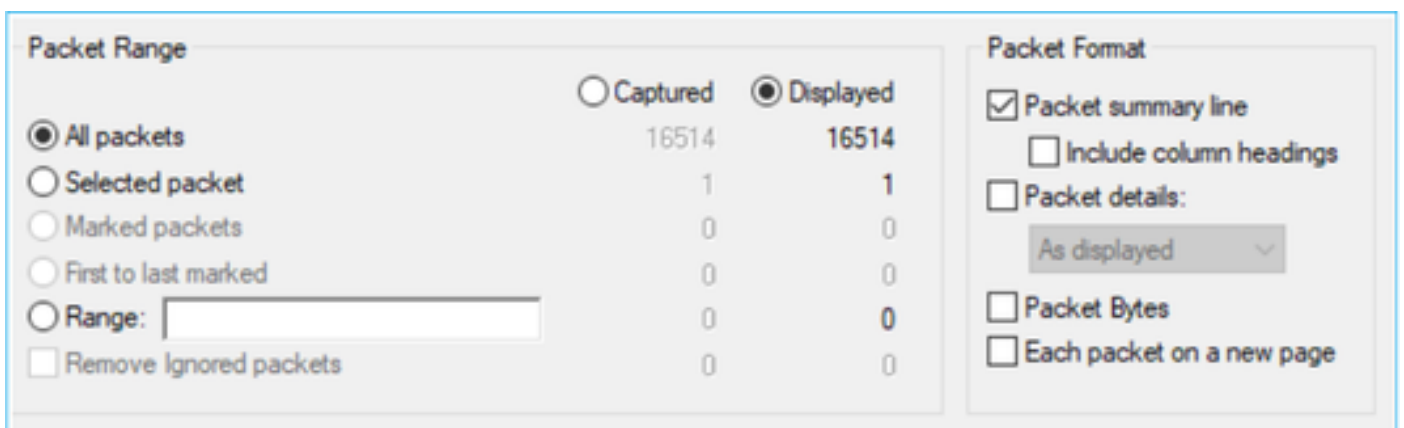
Il risultato:

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

Passaggio 3. Create una versione di testo dell'acquisizione (File > Esporta dismissioni pacchetti > Come testo normale...), come mostrato nell'immagine:



Deselezionare le opzioni Includi intestazioni di colonna e Dettagli pacchetto per esportare solo i valori del campo visualizzato, come mostrato nell'immagine:

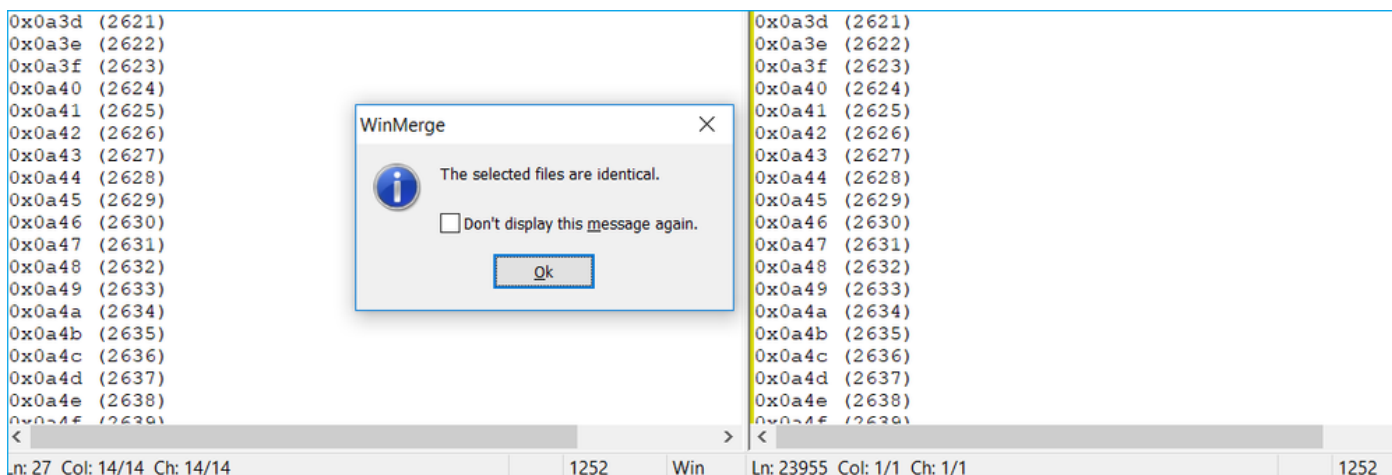


Passaggio 4. Ordinare i pacchetti nei file. A tale scopo, è possibile utilizzare il comando Linux sort:

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

Passaggio 5. Usare uno strumento di confronto del testo (ad esempio, WinMerge) o il comando

Linux diff per trovare le differenze tra le due clip.



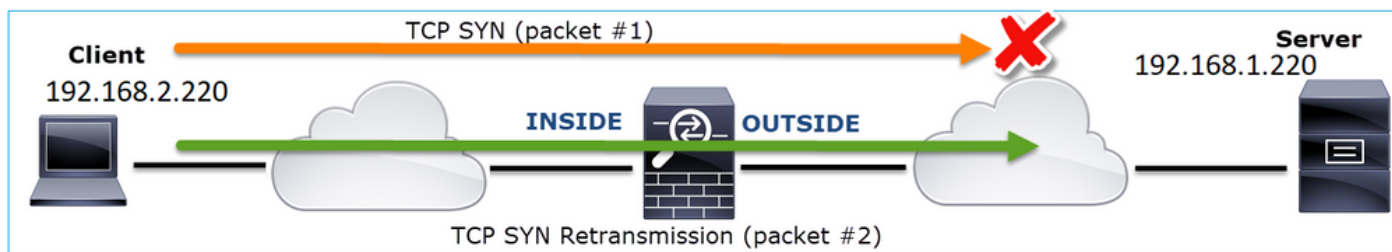
In questo caso, l'acquisizione CAPI e CAPO per il traffico di dati FTP è identica. Ciò dimostra che la perdita del pacchetto non è stata causata dal firewall.

Identificare la perdita di pacchetti a monte e a valle.

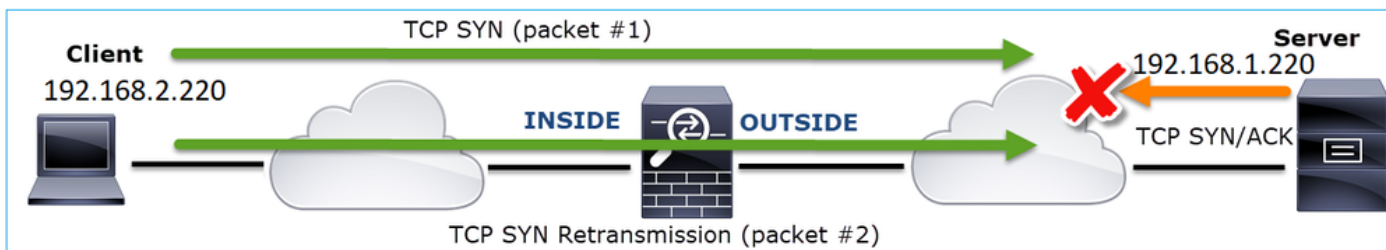
No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-16 16:13:44.169516	192.168.2.220	192.168.1.220	TCP	74	54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=1
2	2019-10-16 16:13:45.196090	192.168.2.220	192.168.1.220	TCP	74	[TCP Retransmission] 54494 → 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=3577291511 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=3577291511 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=3577291510
12	2019-10-16 16:13:47.490407	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
13	2019-10-16 16:13:47.490819	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224321904 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
14	2019-10-16 16:13:47.490880	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224324400 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
15	2019-10-16 16:13:47.490956	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224325648 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
16	2019-10-16 16:13:47.491246	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415

Considerazioni principali:

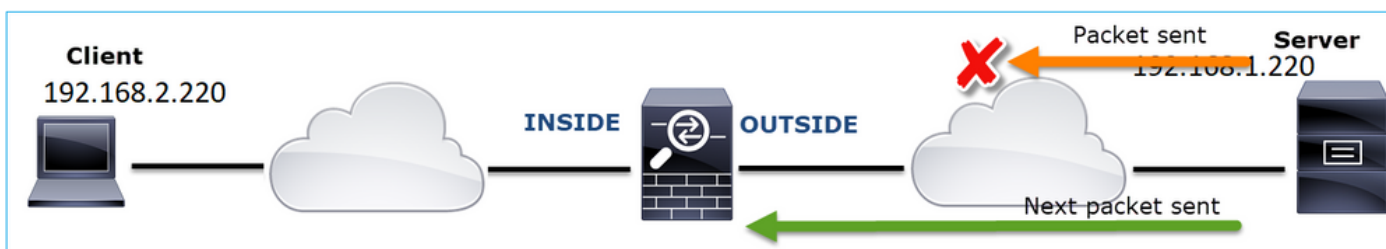
1. Questo pacchetto è una ritrasmissione TCP. In particolare, si tratta di un pacchetto TCP SYN inviato dal client al server per i dati FTP in modalità passiva. Poiché il client invia nuovamente il pacchetto e si può vedere il SYN iniziale (pacchetto 1), il pacchetto è stato perso a monte del firewall.



In questo caso, è possibile che il pacchetto SYN sia arrivato al server, ma che il pacchetto SYN/ACK sia stato perso durante il ritorno:



2. Il server invia un pacchetto e Wireshark identifica che il segmento precedente non è stato rilevato/acquisito. Poiché il pacchetto non acquisito è stato inviato dal server al client e non è stato rilevato nell'acquisizione del firewall, il pacchetto è stato perso tra il server e il firewall.



Ciò indica che esiste una perdita di pacchetti tra il server FTP e il firewall.

Azione 2. Acquisisci Altre Clip.

Acquisire altre clip insieme a quelle sugli endpoint. Provare ad applicare il metodo divide and conquer per isolare ulteriormente il segmento problematico che causa la perdita del pacchetto.

No.	Time	Source	Destination	Protocol	Length	Info
155	2019-10-16 16:13:51.749845	192.168.1.220	192.168.2.220	FTP-DAL	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-DAL	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-DAL	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-DAL	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-DAL	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-DAL	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-DAL	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-DAL	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)

Considerazioni principali:

1. Il ricevitore (in questo caso il client FTP) tiene traccia dei numeri di sequenza TCP in arrivo. Se rileva che un pacchetto è stato perso (un numero di sequenza previsto è stato ignorato), genera un pacchetto ACK con il numero di sequenza previsto ACK='ignorato'. In questo esempio, Ack=2224386800.

2. Il Dup ACK attiva una ritrasmissione rapida TCP (ritrasmissione entro 20 msec dopo la ricezione di un Duplicate ACK).

Cosa significano gli ACK duplicati?

- Alcuni ACK duplicati, ma nessuna ritrasmissione effettiva, indicano che è più probabile che vi siano pacchetti non in ordine.
- La presenza di ACK duplicati, seguiti da ritrasmissioni effettive, indica che si è verificata una perdita di pacchetti.

Azione 3. Calcolare il tempo di elaborazione del firewall per i pacchetti di transito.

Applicare la stessa acquisizione su 2 interfacce diverse:

```
<#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
```

Esportazione dell'acquisizione controllo della differenza di tempo tra i pacchetti in entrata e in uscita

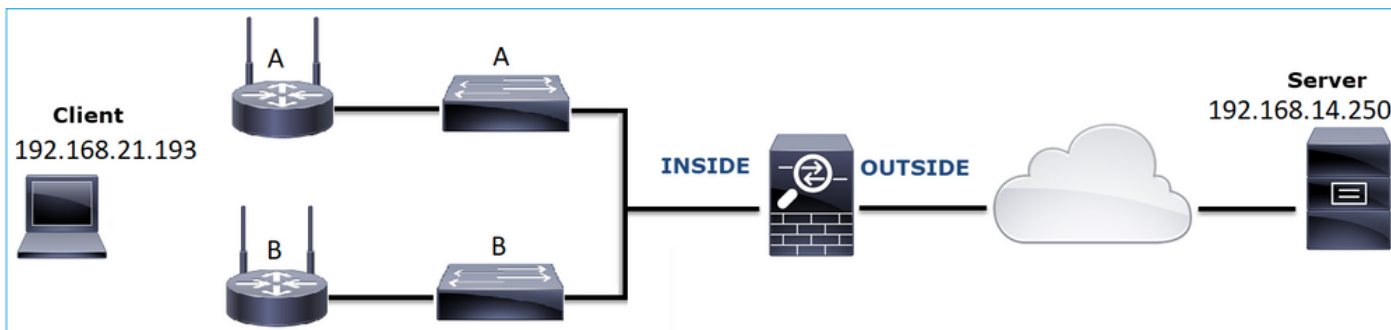
Caso 7. Problema di connettività TCP (pacchetti danneggiati)

Descrizione del problema:

Il client wireless (192.168.21.193) tenta di connettersi a un server di destinazione (192.168.14.250 - HTTP) e vi sono due scenari diversi:

- Quando il client si connette al punto di accesso (punto di accesso) 'A', la connessione HTTP non funziona.
- Quando il client si connette al punto di accesso 'B', la connessione HTTP funziona.

Nell'immagine è illustrata la topologia:



Flusso interessato:

Src IP: 192.168.21.193

Dst IP: 192.168.14.250

Protocollo: TCP 80

Analisi acquisizione

Abilita acquisizioni sul motore LINA 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
```

Clip - Scenario funzionale:

Di base, è sempre molto utile disporre di acquisizioni da uno scenario riconosciuto valido.

Nell'immagine viene mostrata l'acquisizione effettuata sull'interfaccia 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

L'immagine mostra l'acquisizione effettuata sull'interfaccia NGFW OUTSIDE.

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

Considerazioni principali:

1. Le due clip sono quasi identiche (si consideri la randomizzazione ISN).
2. Non ci sono indicazioni di una perdita di pacchetti.
3. Nessun pacchetto non in ordine
4. Sono disponibili 3 richieste HTTP GET. Al primo viene assegnato un 404 "Non trovato", al secondo un "OK" di 200 e al terzo un messaggio di reindirizzamento "Non modificato" di 304.

Acquisizioni - Scenario noto con errori:

Il contenuto CAPI (Ingress Capture).

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=8675756125 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=8675756125 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=8675756125 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=8675756125 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 15:33:35.126846	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

Considerazioni principali:

1. Handshake TCP a 3 vie.
2. Ci sono ritrasmissioni TCP e indicazioni di una perdita di pacchetto.
3. Un pacchetto (TCP ACK) viene identificato da Wireshark come non valido.

L'immagine mostra il contenuto della cattura in uscita (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	HTTP	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

Considerazioni principali:

Le due clip sono quasi identiche (si consideri la randomizzazione ISN):

1. Handshake TCP a 3 vie.
2. Ci sono ritrasmissioni TCP e indicazioni di una perdita di pacchetto.
3. Un pacchetto (TCP ACK) viene identificato da Wireshark come non valido.

Controllare il pacchetto in formato non valido:

The screenshot shows a Wireshark capture of three TCP packets. Packet 3 is identified as a 'Malformed Packet' (1). The packet details pane shows it is a Transmission Control Protocol (ACK) packet with a length of 2 bytes (2). The payload is 'Malformed Packet: Tunnel Socket' (1). The expert info pane shows the error: 'Malformed Packet (Exception occurred)' (1). The packet bytes pane shows the hex and ASCII representation of the packet, with the last two bytes being '00 00' (4).

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909193	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 15:33:31.909849	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=867575959 Ack=4231766829 Win=64240 Len=0 MSS=1380 SACK_PERM=1
3	2013-08-08 15:33:31.913267	192.168.21.193	192.168.14.250	TCP	60	3072 → 80 [ACK] Seq=4231766829 Ack=867575960 Win=65535 Len=2[Malformed Packet]

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

```
0000 58 8d 09 61 cc 9b ec 1a 59 63 90 f3 81 00 00 14  X..a....Yc.....
0010 08 00 45 00 00 2a 7f 1d 40 00 80 06 d5 a4 c0 a8  ..E:.*..@.....
0020 15 c1 c0 a8 0e fa 0c 00 00 50 fc 3b a7 7d 33 b6  .....P:;--3-
0030 28 98 50 10 ff ff 01 bf 00 00 00 00             (-P.....-..)
```

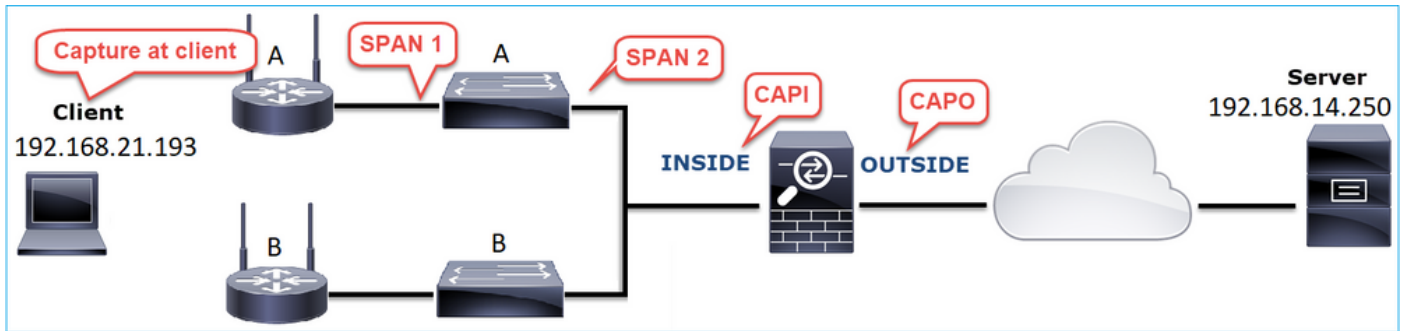
Considerazioni principali:

1. Il pacchetto viene identificato come non valido da Wireshark.
2. Ha una lunghezza di 2 byte.
3. Il payload TCP è di 2 byte.
4. Il carico utile è di 4 zeri in più (00 00).

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Acquisisci altre clip. Includere le acquisizioni negli endpoint e, se possibile, provare ad applicare il metodo divide and conquista per isolare l'origine del danneggiamento del pacchetto, ad esempio:

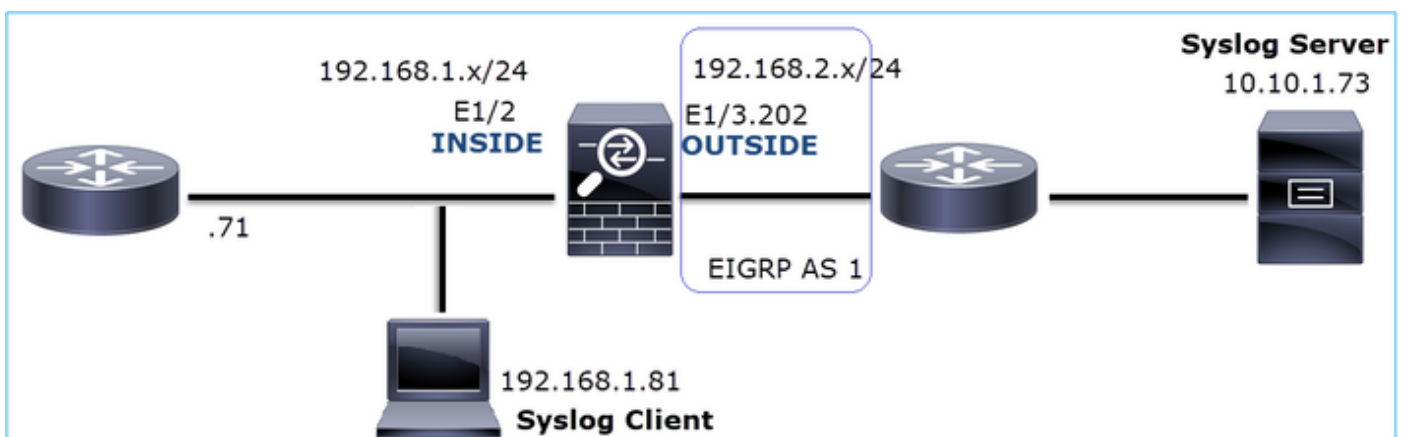


In questo caso, i 2 byte aggiuntivi sono stati aggiunti dal driver dell'interfaccia 'A' dello switch e la soluzione è stata quella di sostituire lo switch che causa il danneggiamento.

Caso 8. Problema di connettività UDP (pacchetti mancanti)

Descrizione del problema: i messaggi Syslog (UDP 514) non vengono visualizzati sul server Syslog di destinazione.

Nell'immagine è illustrata la topologia:



Flusso interessato:

Src IP: 192.168.1.81

Dst IP: 10.10.1.73

Protocollo: UDP 514

Analisi acquisizione

Abilita acquisizioni sul motore LINA 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
```

Le clip FTD non mostrano pacchetti:

```
<#root>
```

```
firepower#
```

```
show capture
```

```
capture CAPI type raw-data trace interface INSIDE [Capturing - 0 bytes]
  match udp host 192.168.1.81 host 10.10.1.73 eq syslog
capture CAPO type raw-data interface OUTSIDE [Capturing - 0 bytes]
  match udp host 192.168.1.81 host 10.10.1.73 eq syslog
```

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Controllare la tabella di connessione FTD.

Per verificare una connessione specifica, è possibile utilizzare la sintassi seguente:

```
<#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
```

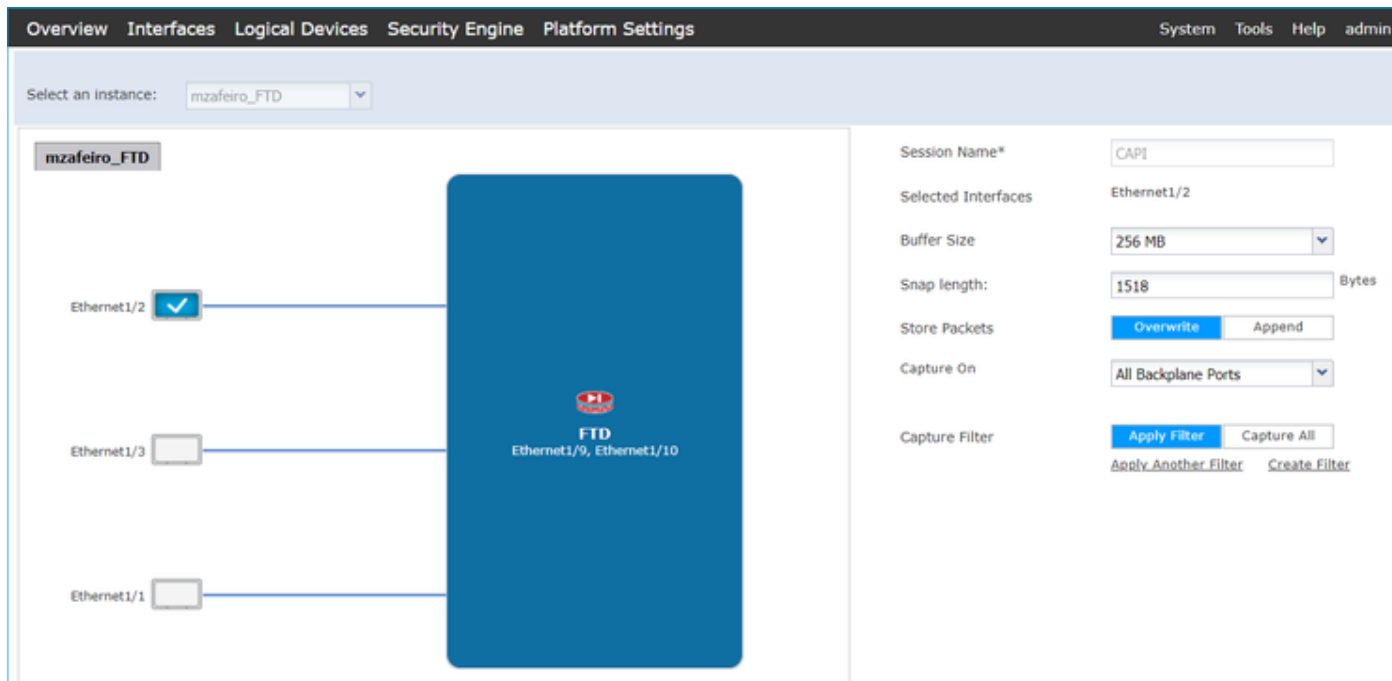
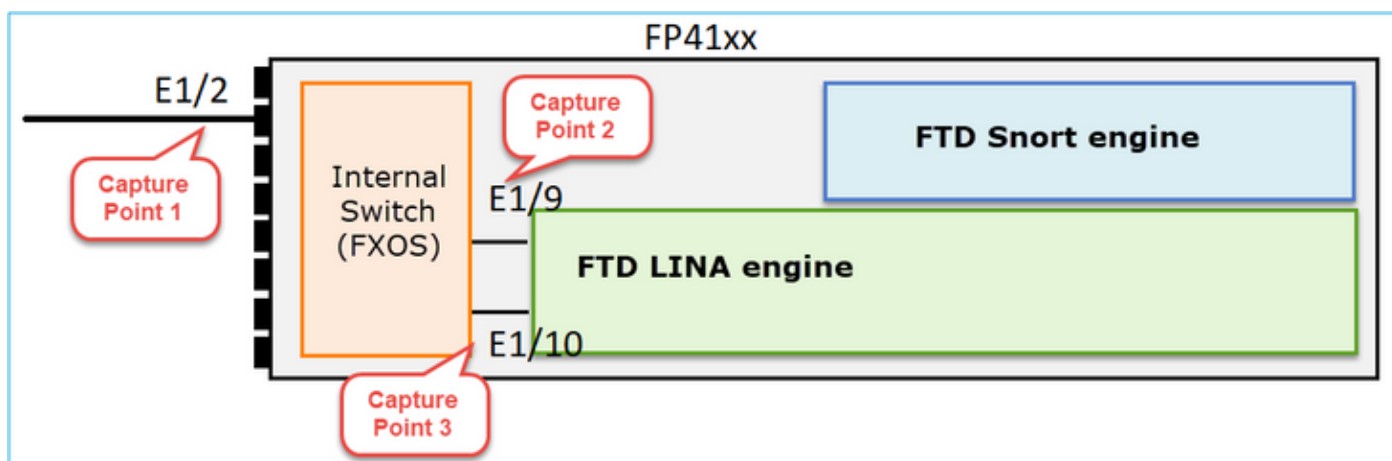
```
N1
```

Considerazioni principali:

1. Le interfacce in entrata e in uscita sono le stesse (inversione a U).
2. Il numero di byte ha un valore molto grande (~5 GByte).
3. Il flag "o" indica l'offload del flusso (flusso accelerato HW). Questo è il motivo per cui le clip FTD non mostrano alcun pacchetto. Flow offload è supportato solo sulle piattaforme 41xx e 93xx. In questo caso, il dispositivo è un 41xx.


Azione 2. Acquisizione a livello di chassis.

Collegarsi al gestore dello chassis Firepower e abilitare l'acquisizione sull'interfaccia in entrata (in questo caso E1/2) e sulle interfacce del backplane (E1/9 e E1/10), come mostrato nell'immagine:



Dopo alcuni secondi:

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

 Suggestione: in Wireshark escludere i pacchetti con tag VN per eliminare la duplicazione dei pacchetti a livello di interfaccia fisica

Prima:

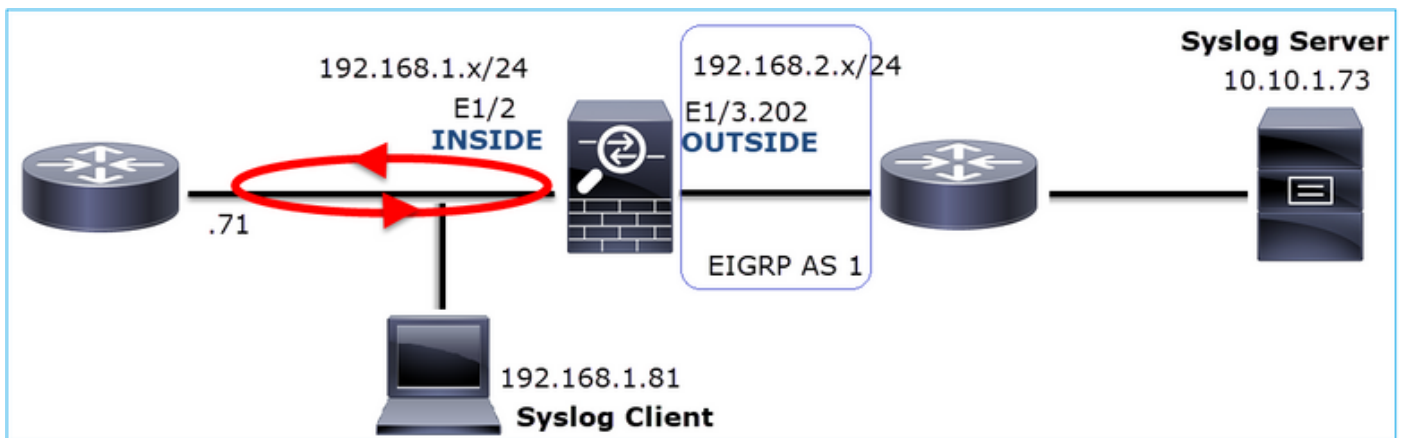
No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
2	0.0000	Cisco_61:5a:9c	Spanning-tree-(f...	STP	64	RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
3	0.0532	Vmware_85:4f:ca	Broadcast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

Dopo:

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.INFO: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
1362	0.000002728	192.168.1.81	10.10.1.73	Syslog	200	254	LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1364	0.000499914	192.168.1.81	10.10.1.73	Syslog	131	251	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1366	0.000697761	192.168.1.81	10.10.1.73	Syslog	147	248	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1368	0.000169137	192.168.1.81	10.10.1.73	Syslog	195	254	LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
1370	0.000433196	192.168.1.81	10.10.1.73	Syslog	151	254	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
1372	0.000498718	192.168.1.81	10.10.1.73	Syslog	200	253	LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1374	0.000002849	192.168.1.81	10.10.1.73	Syslog	131	250	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
1376	0.000596345	192.168.1.81	10.10.1.73	Syslog	147	247	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
1378	0.000600157	192.168.1.81	10.10.1.73	Syslog	195	253	LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
1380	0.000002772	192.168.1.81	10.10.1.73	Syslog	151	253	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
1382	0.000600947	192.168.1.81	10.10.1.73	Syslog	200	252	LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
1384	0.000498808	192.168.1.81	10.10.1.73	Syslog	131	249	LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n

Considerazioni principali:

1. Viene applicato un filtro di visualizzazione per rimuovere i duplicati dei pacchetti e visualizzare solo i syslog.
2. La differenza tra i pacchetti è al livello del microsecondo. Ciò indica una velocità di trasmissione dei pacchetti molto elevata.
3. Il valore TTL (Time to Live) diminuisce continuamente. Ciò indica un loop di pacchetto.



Azione 3. Usare packet-tracer.

Poiché i pacchetti non attraversano il motore LINA del firewall, non è possibile eseguire una traccia in tempo reale (acquisizione con traccia), ma è possibile tracciare un pacchetto emulato 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
```

Azione 4. Confermare il ciclo FTD.

Controllare la tabella di routing del firewall per verificare se sono presenti problemi di routing:

```
<#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
```

Considerazioni principali:

1. Il percorso punta verso l'interfaccia di uscita corretta.
2. Il percorso è stato appreso qualche minuto fa (0:02:37).

Azione 5. Confermare il tempo di attività della connessione.

Controllare il tempo di attività della connessione per verificare quando è stata stabilita la connessione:

```
<#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 chiave:

1. La connessione è stata stabilita circa 4 minuti fa (prima dell'installazione del percorso EIGRP nella tabella di routing)

Azione 6. Cancellare la connessione stabilita.

In questo caso, i pacchetti corrispondono a una connessione stabilita e vengono instradati a un'interfaccia di uscita errata; ciò provoca un loop. Ciò è dovuto all'ordine delle operazioni del firewall:

1. Ricerca di connessioni stabilita (ha priorità rispetto alla ricerca nella tabella di routing globale).
2. Ricerca NAT (Network Address Translation) - La fase UN-NAT (NAT di destinazione) ha la precedenza sulla ricerca PBR e route.
3. Policy-Based Routing (PBR)
4. Ricerca nella tabella di routing globale

Poiché la connessione non scade mai (il client Syslog invia continuamente i pacchetti mentre il timeout di inattività della connessione UDP è di 2 minuti), è necessario cancellare manualmente la connessione:

```
<#root>
```

```
firepower#
```

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

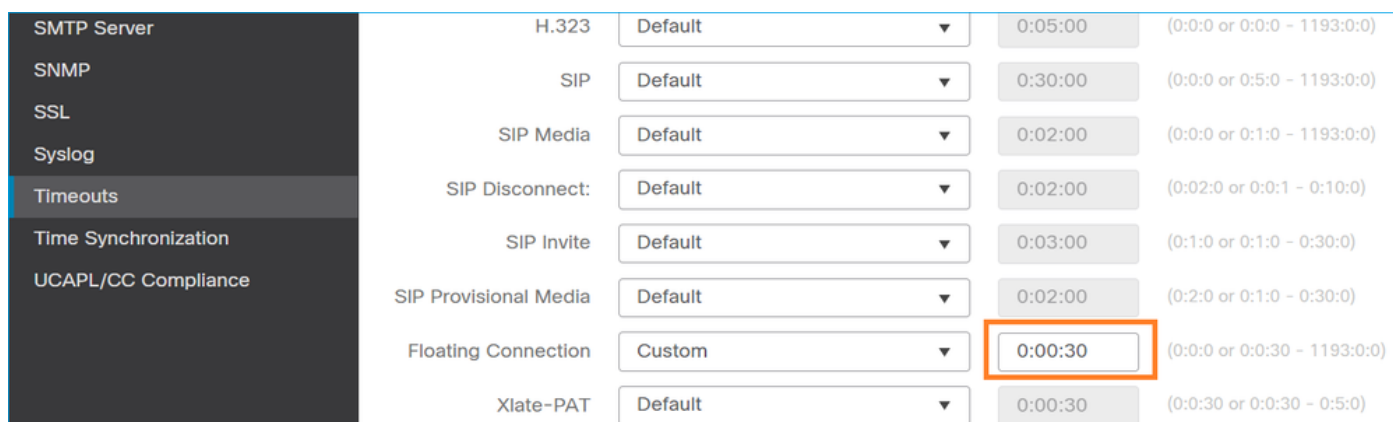
```
1 connection(s) deleted.
```

Verificare che sia stata stabilita una nuova connessione:

```
<#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
```

Azione 7. Configurare il timeout della connessione mobile.

Questa è la soluzione ideale per risolvere il problema ed evitare un routing non ottimale, soprattutto per i flussi UDP. Passare a Dispositivi > Impostazioni piattaforma > Timeout e impostare il valore:



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)

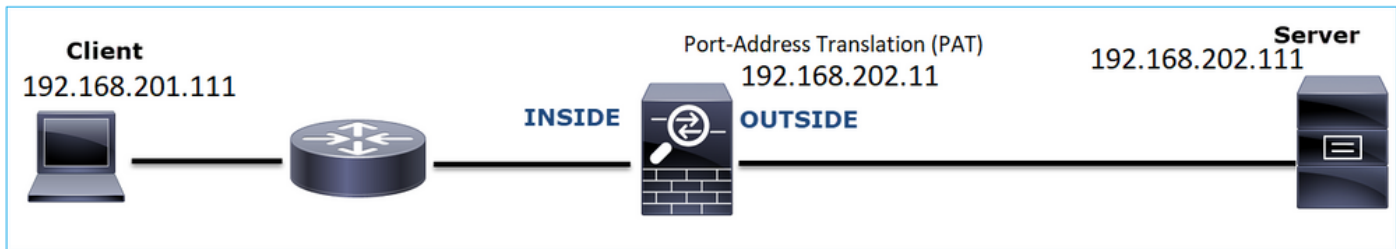
Per ulteriori informazioni sul timeout della conn mobile, vedere la Guida di riferimento per i comandi:

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

Caso 9. Problema di connettività HTTPS (scenario 1)

Descrizione del problema: impossibile stabilire la comunicazione HTTPS tra il client 192.168.201.105 e il server 192.168.202.101

Nell'immagine è illustrata la topologia:



Flusso interessato:

Src IP: 192.168.201.111

Indirizzo IP: 192.168.202.111

Protocollo: TCP 443 (HTTPS)

Analisi acquisizione

Abilita acquisizioni sul motore LINA FTD:

L'indirizzo IP utilizzato nell'acquisizione OUTSIDE è diverso a causa della configurazione Port-Address Translation.

```
<#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
```

Nell'immagine viene mostrata l'acquisizione effettuata sull'interfaccia 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	TLsv1	326	0x2f33 (12083)	Client Hello
42	2018-02-01 10:39:35.252352	192.168.202.111	192.168.201.111	TCP	70	0xefb4 (61364)	443 → 6666 [ACK] Seq=4086514532 Ack=2034865888 Win=8192 Len=0 TSval=3119615816 TSecr=192658174
43	2018-02-01 10:40:05.317320	192.168.202.111	192.168.201.111	TCP	70	0xd8c3 (55491)	443 → 6666 [RST] Seq=4086514532 Win=8192 Len=0 TSval=3119645908 TSecr=0

Considerazioni principali:

1. Handshake TCP a 3 vie.
2. Avvio della negoziazione SSL. Il client invia un messaggio Hello al client.
3. Al client è stato inviato un ACK TCP.
4. RST TCP inviato al client.

L'immagine mostra l'acquisizione effettuata sull'interfaccia NGFW OUTSIDE.

No.	Time	Source	Destination	Protocol	Length	Identification	Info
33	2018-02-01 10:39:35.188192	192.168.202.11	192.168.202.111	TCP	78	0x2f31 (12081)	15880 → 443 [SYN] Seq=2486930707 Win=29200 Len=0 MSS=1380 SACK_PERM=1 TSval=192658158 TSecr=0 WS=128
34	2018-02-01 10:39:35.188527	192.168.202.111	192.168.202.11	TCP	78	0x0000 (0)	443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=0
35	2018-02-01 10:39:35.189214	192.168.202.11	192.168.202.111	TCP	70	0x2f32 (12082)	15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Win=29312 Len=0 TSval=192658158 TSecr=3119615816
36	2018-02-01 10:39:35.252397	192.168.202.11	192.168.202.111	TLSv1	257	0xcd56 (52534)	Client Hello
37	2018-02-01 10:39:37.274430	192.168.202.11	192.168.202.111	TCP	257	0x0995 (41265)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192660198 TSecr=0
38	2018-02-01 10:39:41.297332	192.168.202.11	192.168.202.111	TCP	257	0x88af (34091)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192664224 TSecr=0
39	2018-02-01 10:39:49.309569	192.168.202.11	192.168.202.111	TCP	257	0xf68a (63114)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=8192 Len=187 TSval=192672244 TSecr=0
40	2018-02-01 10:40:05.317305	192.168.202.11	192.168.202.111	TCP	78	0xd621 (54817)	15880 → 443 [RST] Seq=2486930895 Win=0 Len=0 TSval=192688266 TSecr=0
41	2018-02-01 10:40:06.790700	192.168.202.111	192.168.202.11	TCP	78	0x0000 (0)	[TCP Retransmission] 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3119615816 TSecr=0

Considerazioni principali:

1. Handshake TCP a 3 vie.
2. Avvio della negoziazione SSL. Il client invia un messaggio Hello al client.
3. Sono presenti ritrasmissioni TCP inviate dal firewall al server.
4. RST TCP inviato al server.

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

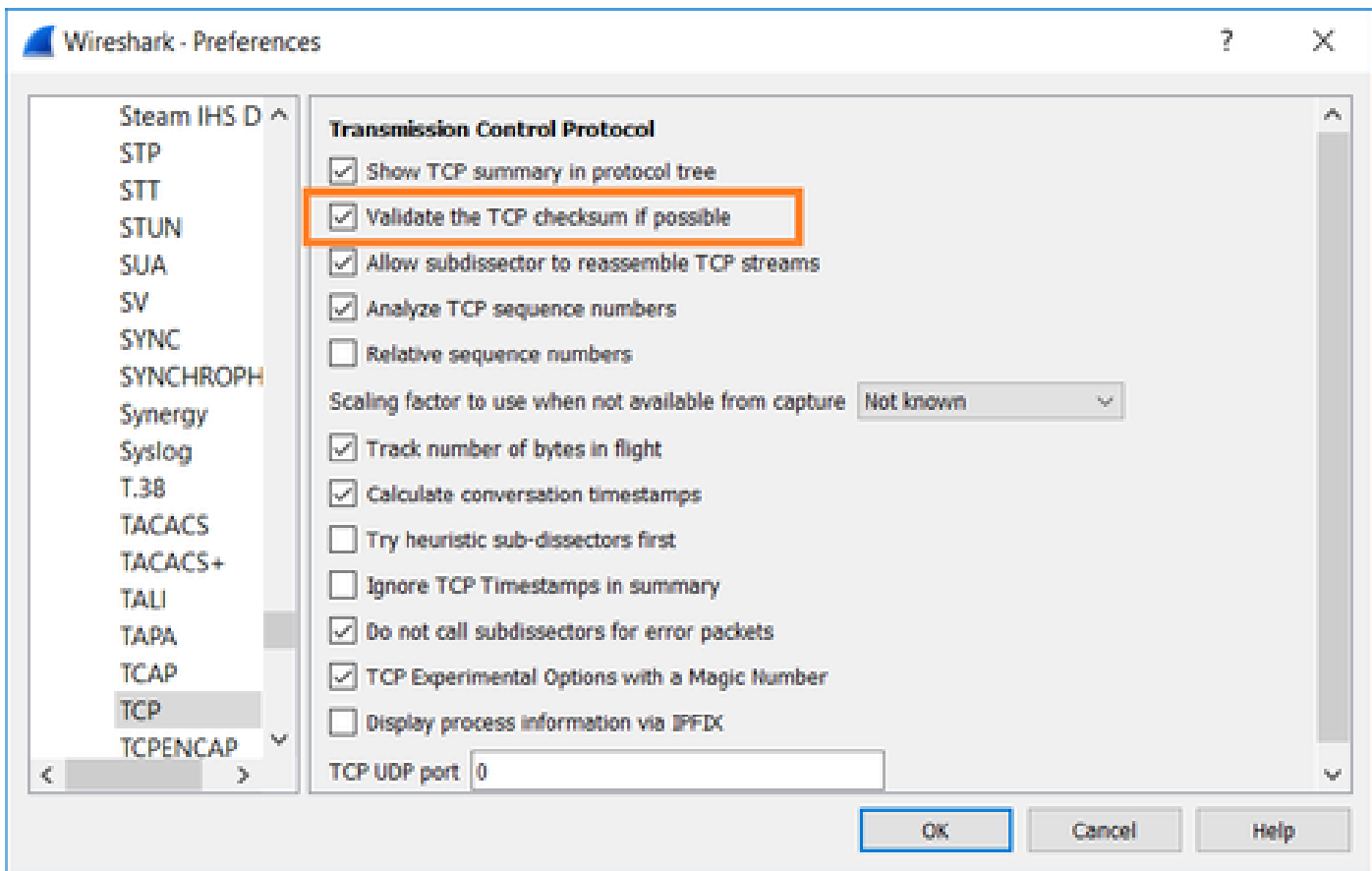
Azione 1. Acquisisci altre clip.

Un'acquisizione effettuata sul server rivela che il server ha ricevuto gli Helper del client TLS con checksum TCP corrotti e li scarta in modo invisibile all'utente (non esiste TCP RST o alcun altro pacchetto di risposta verso il client):

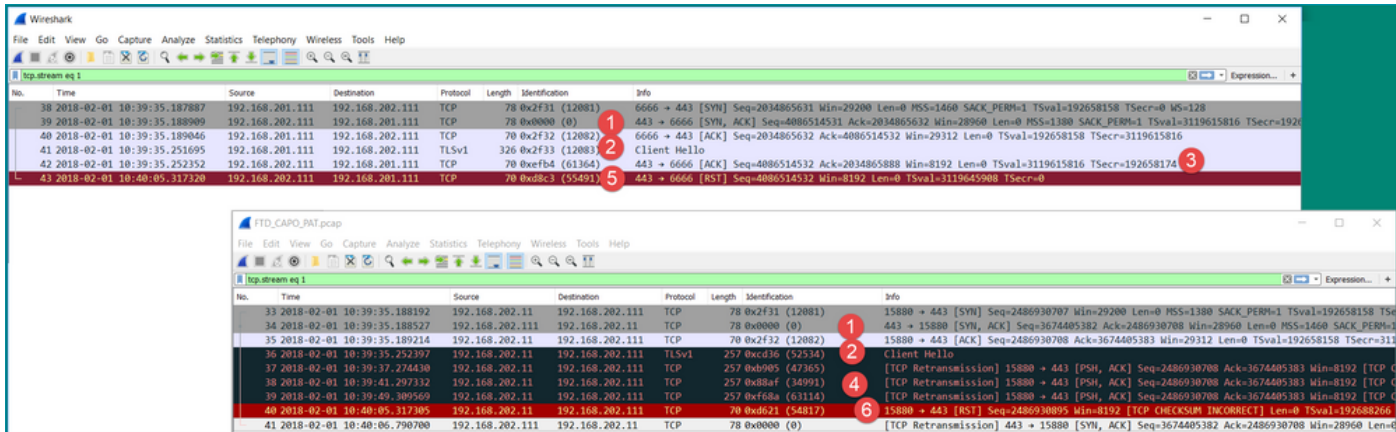
```
21:26:27.133677 IP (tos 0x0, ttl 64, id 52534, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x0c65 (incorrect -> 0x3063), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192658174 ecr 3119615816], length 187
21:26:29.155652 IP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

Quando si mette tutto insieme:

In questo caso, per comprenderlo, è necessario abilitare su Wireshark l'opzione Validate the TCP checksum (Convalida checksum TCP se possibile). Passare a Modifica > Preferenze > Protocolli > TCP, come mostrato nell'immagine.



In questo caso, è utile mettere le clip una accanto all'altra per avere una visione completa:



Considerazioni principali:

1. Handshake TCP a 3 vie. Gli ID IP sono gli stessi. Ciò significa che il flusso non è stato inviato tramite proxy dal firewall.
2. Il client TLS Hello viene dal client con ID IP 12083. Il pacchetto viene inoltrato dal firewall (in questo caso, il firewall è stato configurato con i criteri di decrittografia TLS) e l'ID IP viene modificato in 52534. Inoltre, il checksum TCP del pacchetto viene danneggiato (a causa di un difetto del software che in seguito è stato risolto).
3. Il firewall è in modalità Proxy TCP e invia un ACK al client (che falsifica il server).

```

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

```

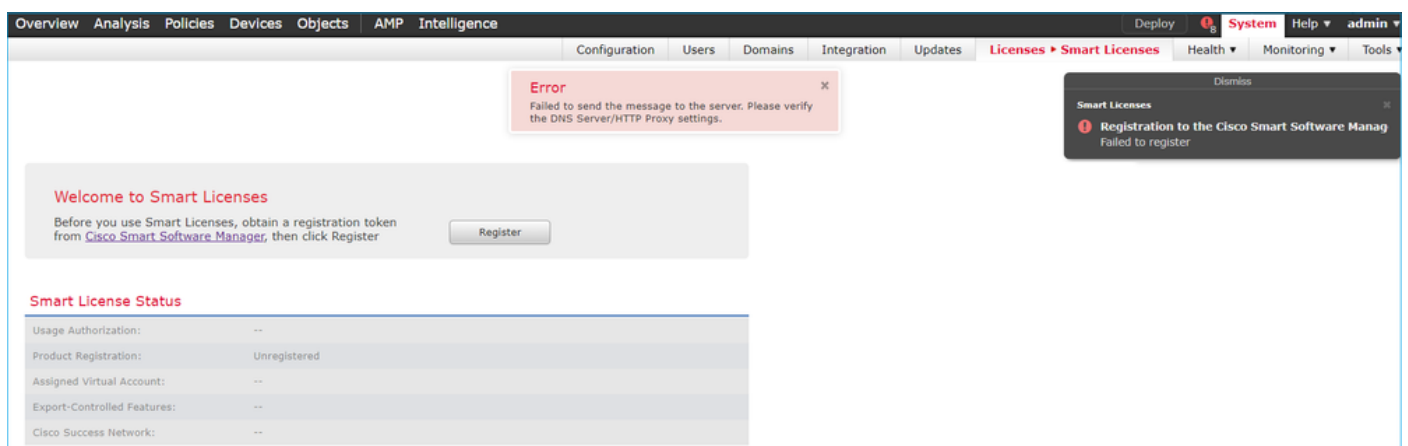
4. Il firewall non riceve alcun pacchetto TCP ACK dal server e trasmette nuovamente il messaggio Hello del client TLS. Anche in questo caso, la causa è la modalità TCP Proxy attivata dal firewall.
5. Dopo circa 30 secondi, il firewall si disattiva e invia un RST TCP al client.
6. Il firewall invia una richiesta RST TCP verso il server.

Per riferimento:

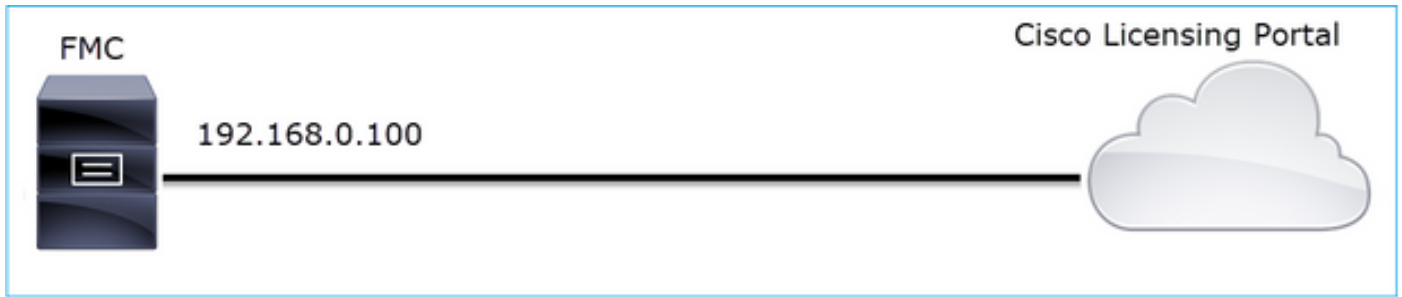
[Elaborazione handshake Firepower TLS/SSL](#)

Caso 10. Problema di connettività HTTPS (scenario 2)

Descrizione del problema: la registrazione della licenza Smart License di FMC non è riuscita.



Nell'immagine è illustrata la topologia:



Flusso interessato:

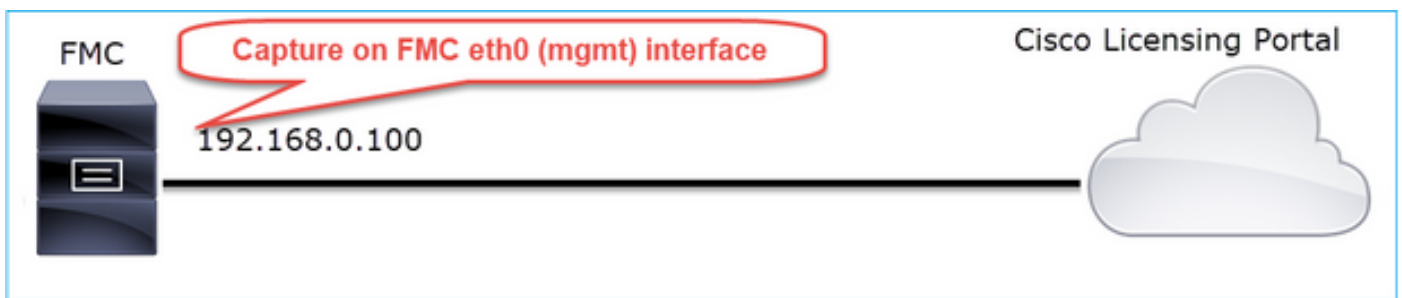
Src IP: 192.168.0.100

Dst: tools.cisco.com

Protocollo: TCP 443 (HTTPS)

Analisi acquisizione

Abilitare l'acquisizione sull'interfaccia di gestione FMC:



Riprovare a eseguire la registrazione. Quando viene visualizzato il messaggio Error (Errore), premere CTRL-C per interrompere l'acquisizione:

```
<#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
```

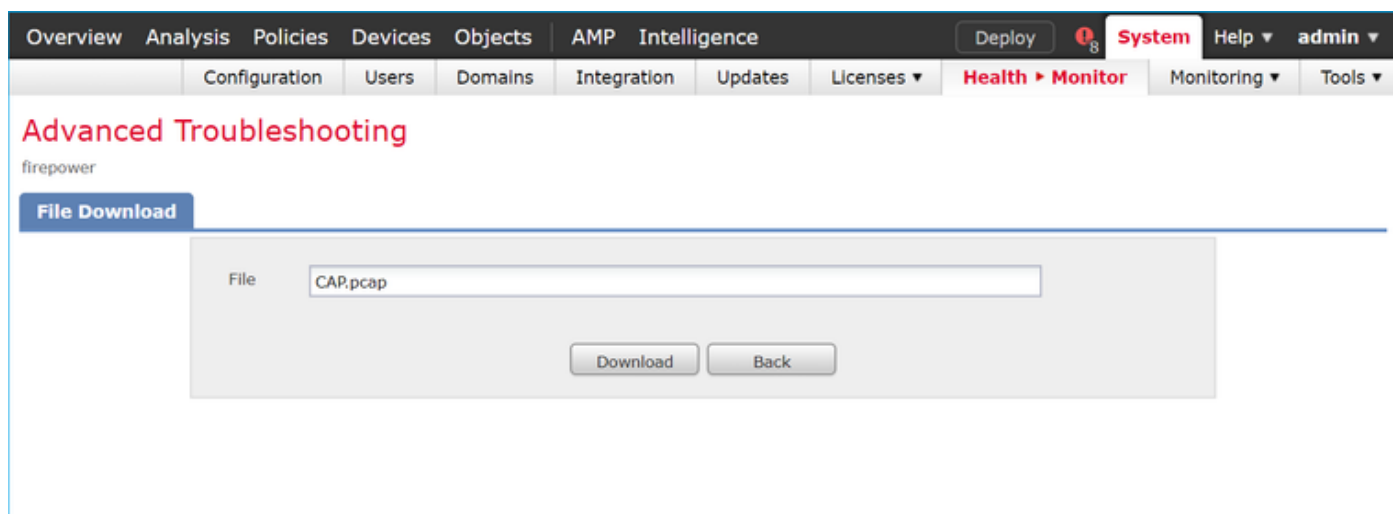
```
<- CTRL-C
```

```
264 packets received by filter
```

```
0 packets dropped by kernel
```


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


Raccogliere l'acquisizione dal FMC (Sistema > Integrità > Monitor, selezionare il dispositivo e selezionare Advanced Troubleshooting), come mostrato nell'immagine:




L'immagine mostra l'acquisizione della FMC su 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

 Suggerimento: per controllare tutte le nuove sessioni TCP acquisite, utilizzare il filtro di visualizzazione `tcp.flags==0x2` su Wireshark. In questo modo vengono filtrati tutti i pacchetti TCP SYN acquisiti.

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

 Suggerimento: applicare come colonna il campo Nome server dal client SSL Hello.

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

Suggestione: applicare questo filtro di visualizzazione per visualizzare solo i messaggi del client Hello `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: al momento della stesura di questo documento, il portale delle licenze intelligenti (tools.cisco.com) utilizza gli indirizzi IP seguenti: 72.163.4.38, 173.37.145.8

Seguire uno dei flussi TCP (Segui > Flusso TCP), come mostrato nell'immagine.

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

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74		35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16127801 TSecr=0 WS=128
73	2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [SYN, ACK] Seq=2770078884 Ack=2427943532 Win=8190 Len=0 MSS=1330
74	2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427943532 Ack=2770078885 Win=29200 Len=0
75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
76	2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=32768 Len=0
77	2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150		Server Hello
78	2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=29200 Len=0
79	2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1304		443 → 35752 [PSH, ACK] Seq=2770078981 Ack=2427944049 Win=32768 Len=1330 [TCP segment of a reassembled PDU]
80	2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=31920 Len=0
81	2019-10-23 07:45:14.966877	72.163.4.38	192.168.0.100	TLSv1.2	155		Certificate
82	2019-10-23 07:45:14.966885	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=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=31920 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)

Considerazioni principali:

1. Handshake TCP a 3 vie.
2. Il client (FMC) invia un messaggio Hello per il client SSL al portale di Smart Licensing.
3. L'ID sessione SSL è 0. Ciò significa che non si tratta di una sessione ripresa.
4. Il server di destinazione risponde con il messaggio Server Hello, Certificate e Server Hello Done.
5. Il client invia un avviso SSL irreversibile relativo a un'autorità di certificazione sconosciuta.
6. Il client invia una richiesta RST TCP per chiudere la sessione.
7. L'intera durata della sessione TCP (dalla definizione alla chiusura) è stata di circa 0,5 sec.

Selezionare il Certificato server ed espandere il campo autorità emittente per visualizzare il nome comune. In questo caso, il nome comune indica un dispositivo che esegue il comando 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)
                        > RDNSquence item: 1 item (id-at-organizationName=FTD_O)
                        > RDNSquence item: 1 item (id-at-organizationalUnitName=FTD_OU)
                        > RDNSquence item: 1 item (id-at-commonName=FTD4100_MITM)
                  > validity
                  > subject: rdnSequence (0)
                  > subjectPublicKeyInfo
                > extensions: 6 items
  
```

Questa è la figura:

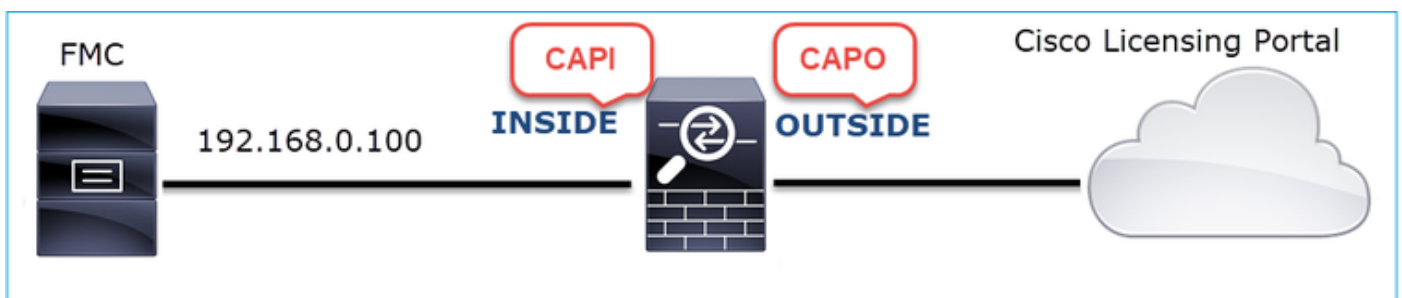


Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Acquisisci altre clip.

Acquisire immagini sul dispositivo firewall di transito:



CAPI mostra:

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=1330
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_MITH,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_MITH)
    validity
  
```

CAPO mostra:

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

```

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

Queste acquisizioni dimostrano che il firewall di transito modifica il certificato server (MITM)

Azione 2. Controllare i registri del dispositivo.

È possibile raccogliere il bundle FMC TS come descritto nel presente documento:

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

In questo caso, il file /dir-archives/var-log/process_stdout.log visualizza messaggi come questo:

```
<#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/
```

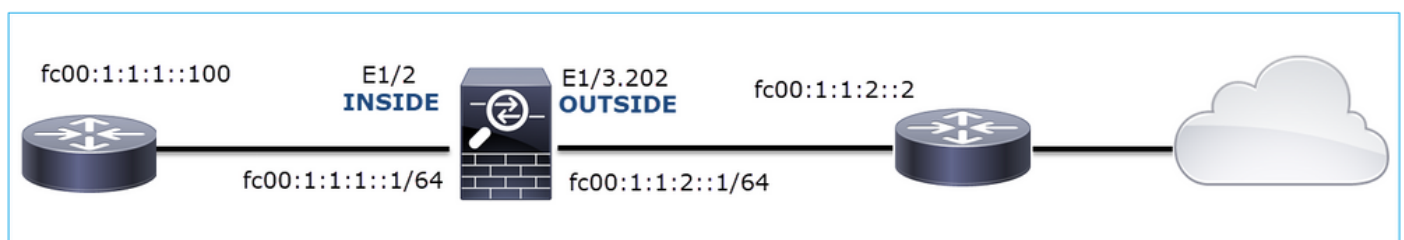
Soluzione consigliata

Disabilitare il servizio MITM per il flusso specifico in modo che FMC possa eseguire correttamente la registrazione nel cloud di Smart Licensing.

Caso 11. Problema di connettività IPv6

Descrizione del problema: gli host interni (dietro l'interfaccia INSIDE del firewall) non sono in grado di comunicare con gli host esterni (dietro l'interfaccia OUTSIDE del firewall).

Nell'immagine è illustrata la topologia:



Flusso interessato:

Src IP: fc00:1:1:1::100

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

Protocollo: qualsiasi

Analisi acquisizione

Abilita le acquisizioni sul motore LINA FTD.

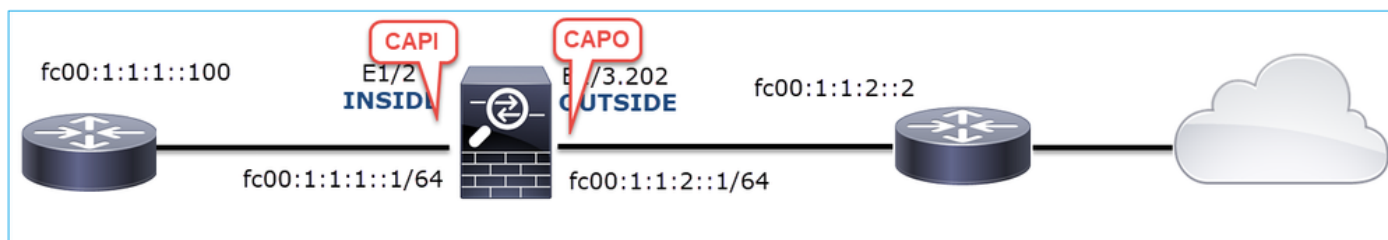
```
<#root>
```

```
firepower#
```

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

```
firepower#
```

capture CAPO int OUTSIDE match ip any6 any6



Acquisizioni - Scenario non funzionale

Le clip sono state acquisite in parallelo con un test di connettività ICMP da IP fc00:1:1:1:10 (router interno) a IP fc00:1:1:2:2 (router upstream).

L'interfaccia INSIDE di acquisizione su firewall 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)

Considerazioni principali:

1. Il router invia un messaggio di richiesta router adiacente IPv6 e richiede l'indirizzo MAC del dispositivo upstream (IP fc00:1:1:1:1).
2. Il firewall risponde con un annuncio router adiacente IPv6.
3. Il router invia una richiesta Echo ICMP.
4. Il firewall invia un messaggio di richiesta router adiacente IPv6 e richiede l'indirizzo MAC del dispositivo downstream (fc00:1:1:1:100).
5. Il router risponde con un annuncio router adiacente IPv6.
6. Il router invia ulteriori richieste echo ICMP IPv6.

L'acquisizione sull'interfaccia ESTERNA 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	118	Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
4	2019-10-24 13:02:09.001815	fc00:1:1:2:2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
5	2019-10-24 13:02:10.025938	fc00:1:1:2:2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
6	2019-10-24 13:02:10.998132	fc00:1:1:1::100	fc00:1:1:2:2	ICMPv6	118	Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
7	2019-10-24 13:02:11.050015	fc00:1:1:2:2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
8	2019-10-24 13:02:12.066082	fe80::4e4e:35ff:fefc:fc:d8	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:fc:d8	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:fc:d8	ICMPv6	90	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fc:d8 from 00:be:75:f6:1d:8e
16	2019-10-24 13:02:17.062512	fe80::4e4e:35ff:fefc:fc:d8	fe80::2be:75ff:fef6:1d8e	ICMPv6	82	Neighbor Advertisement fe80::4e4e:35ff:fefc:fc:d8 (rtr, sol)

Considerazioni principali:

1. Il firewall invia un messaggio di richiesta router adiacente IPv6 che richiede l'indirizzo MAC

- del dispositivo upstream (IP fc00:1:1:2:2).
2. Il router risponde con un annuncio router adiacente IPv6.
 3. Il firewall invia una richiesta echo ICMP IPv6.
 4. Il dispositivo upstream (router fc00:1:1:2:2) invia un messaggio di richiesta router adiacente IPv6 che richiede l'indirizzo MAC dell'indirizzo IPv6 fc00:1:1:1:100.
 5. Il firewall invia una richiesta echo ICMP IPv6 aggiuntiva.
 6. Il router upstream invia un messaggio aggiuntivo di richiesta router adiacente IPv6 che richiede l'indirizzo MAC dell'indirizzo IPv6 fc00:1:1:1:100.

Il punto 4 è molto interessante. Normalmente il router a monte chiede l'indirizzo MAC dell'interfaccia OUTSIDE del firewall (fc00:1:1:2:2), ma invece chiede l'indirizzo fc00:1:1:1:100. Ciò indica una configurazione errata.

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Controllare la tabella adiacente IPv6.

La tabella adiacente IPv6 del firewall è popolata correttamente.

```
<#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
```

Azione 2. Controllare la configurazione IPv6.

Questa è la configurazione 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 configurazione del dispositivo a monte rivela la configurazione errata:

```

<#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

```

Acquisizioni - Scenario funzionale

La modifica della subnet mask (da /48 a /64) ha risolto il problema. Questa è l'acquisizione CAPI nello scenario funzionale.

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 chiave:

1. Il router invia un messaggio di richiesta router adiacente IPv6 che richiede l'indirizzo MAC

- del dispositivo upstream (IP fc00:1:1:1:1).
- 2. Il firewall risponde con un annuncio router adiacente IPv6.
- 3. Il router invia richieste echo ICMP e riceve risposte echo.

Contenuto del 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)

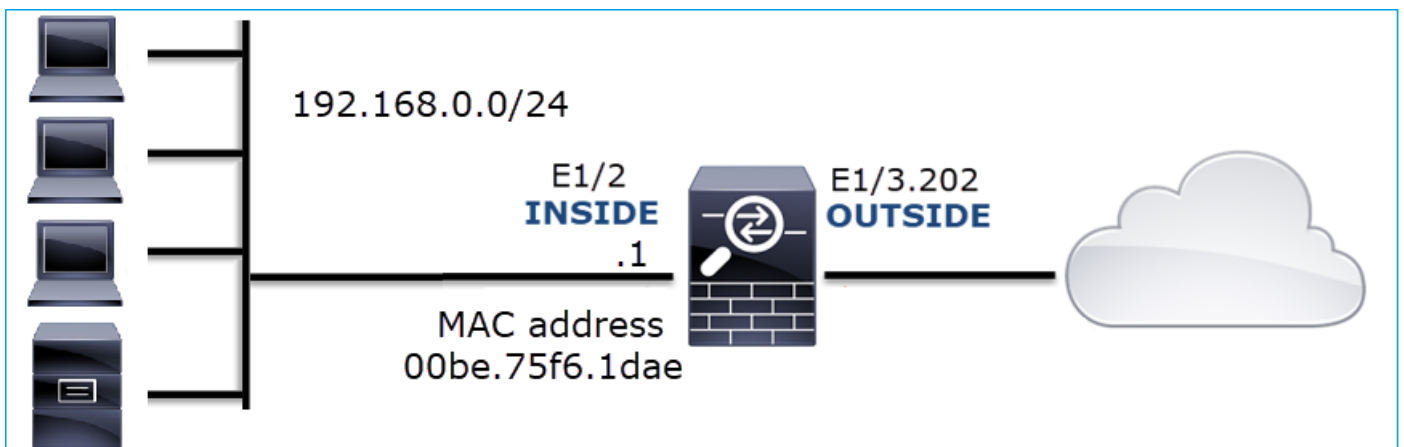
Considerazioni principali:

1. Il firewall invia un messaggio di richiesta router adiacente IPv6 che richiede l'indirizzo MAC del dispositivo upstream (IP fc00:1:1:2:2).
2. Il firewall risponde con un annuncio router adiacente IPv6.
3. Il firewall invia una richiesta echo ICMP.
4. Il router invia un messaggio di richiesta router adiacente IPv6 che richiede l'indirizzo MAC del dispositivo downstream (IP fc00:1:1:1:1).
5. Il firewall risponde con un annuncio router adiacente IPv6.
6. Il firewall invia richieste echo ICMP e riceve risposte echo.

Caso 12. Problema di connettività intermittente (avvelenamento ARP)

Descrizione del problema: gli host interni (192.168.0.x/24) presentano problemi di connettività intermittenti con gli host della stessa subnet

Nell'immagine è illustrata la topologia:



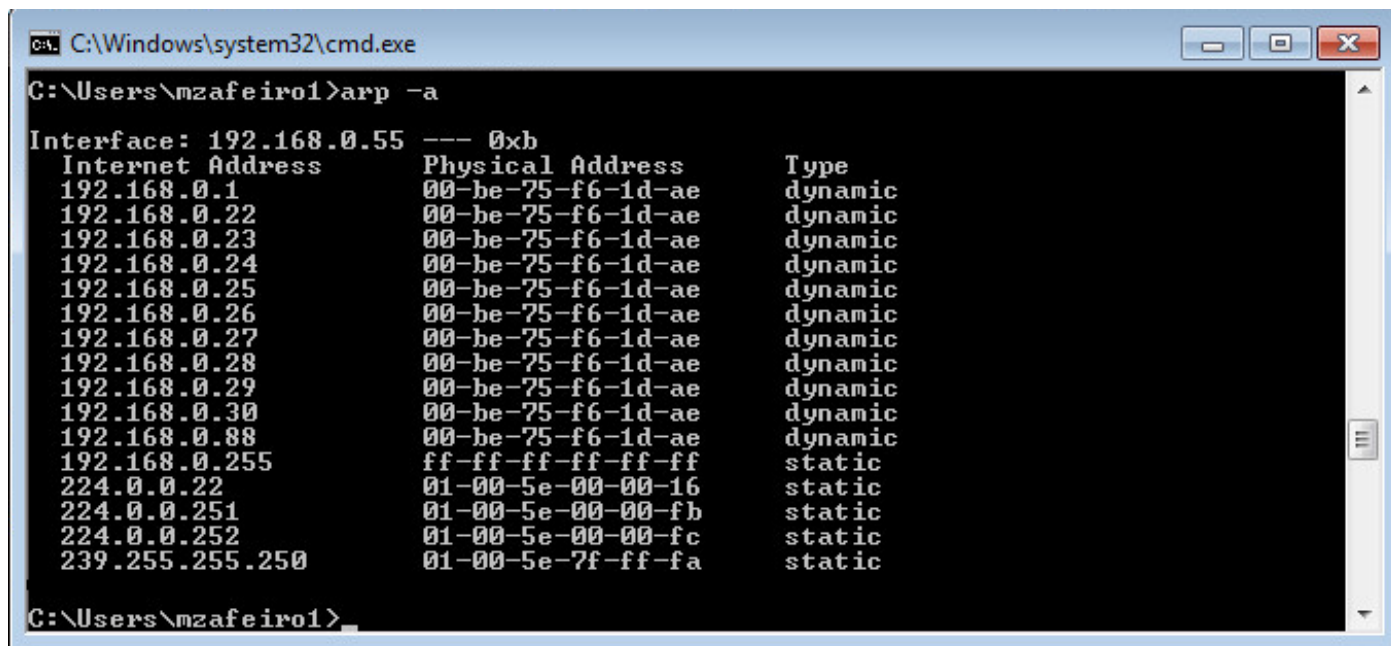
Flusso interessato:

Src IP: 192.168.0.x/24

Dst IP: 192.168.0.x/24

Protocollo: qualsiasi

La cache ARP di un host interno sembra essere avvelenata:



```
C:\Windows\system32\cmd.exe
C:\Users\mzafeiro1>arp -a

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

C:\Users\mzafeiro1>
```

Analisi acquisizione

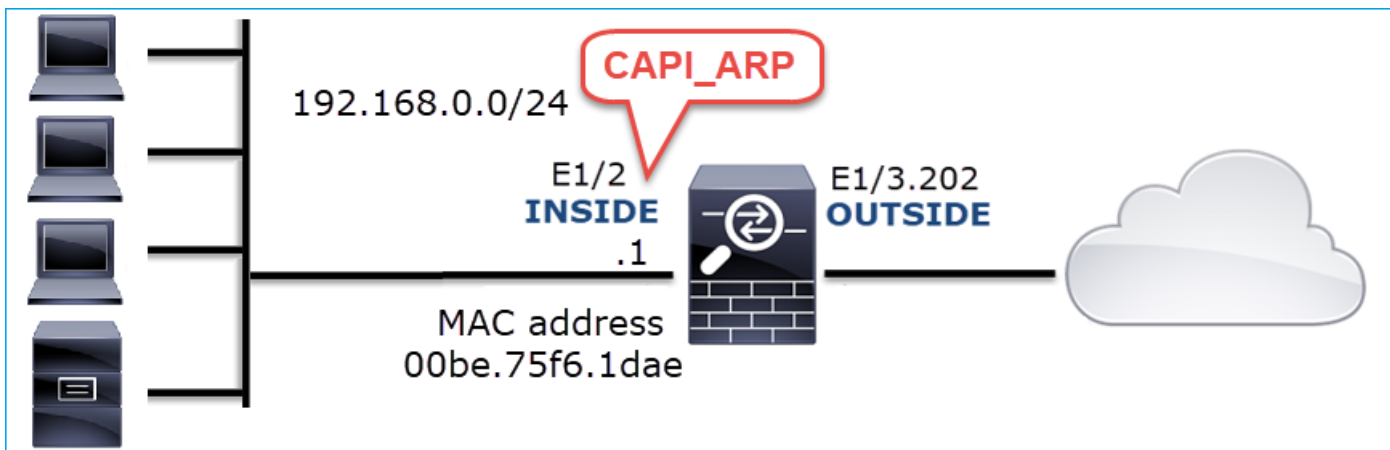
Abilita un'acquisizione sul motore LINA FTD

Questa acquisizione acquisisce solo i pacchetti ARP sull'interfaccia INSIDE:

```
<#root>
```

```
firepower#
```

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



Acquisizioni - Scenario non funzionale:

L'acquisizione sull'interfaccia INSIDE del firewall contiene.

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

Considerazioni principali:

1. Il firewall riceve varie richieste ARP per gli IP all'interno della rete 192.168.0.x/24
2. Il firewall risponde a tutti (proxy-ARP) con il proprio indirizzo MAC

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Controllare la configurazione NAT.

Per quanto riguarda la configurazione NAT, in alcuni casi la parola chiave no-proxy-arp può impedire il comportamento precedente:

```
<#root>
```

```
firepower#
```

```
show run nat
```

```
nat (INSIDE,OUTSIDE) source static NET_1.1.1.0 NET_2.2.2.0 destination static NET_192.168.0.0 NET_4.4.4
```

no-proxy-arp

Azione 2. Disabilitare la funzionalità proxy-arp sull'interfaccia del firewall.

Se la parola chiave "no-proxy-arp" non risolve il problema, provare a disabilitare il proxy ARP sull'interfaccia stessa. Nel caso di FTD, al momento della scrittura, è necessario utilizzare FlexConfig e distribuire il comando (specificare il nome dell'interfaccia appropriato).

```
sysopt noproxyarp INSIDE
```

Caso 13. Identificazione degli identificatori di oggetti (OID) SNMP che causano il blocco della CPU

In questo caso viene mostrato come alcuni OID SNMP per il polling della memoria sono stati identificati come la causa principale dei log della CPU (problema di prestazioni) in base all'analisi delle acquisizioni di pacchetti SNMP versione 3 (SNMPv3).

Descrizione del problema: gli overrun sulle interfacce dati aumentano continuamente. Ulteriori ricerche hanno rivelato che ci sono anche hook della CPU (causati dal processo SNMP) che sono la causa principale dei sovraccarichi dell'interfaccia.

Il passaggio successivo del processo di risoluzione dei problemi consisteva nell'identificare la causa principale dei log della CPU causati dal processo SNMP e, in particolare, restringere l'ambito del problema per identificare gli identificatori di oggetto SNMP (OID) che, se esaminati, potrebbero potenzialmente causare hog della CPU.

Al momento, il motore LINA FTD non fornisce un comando 'show' per gli OID SNMP di cui viene eseguito il polling in tempo reale.

L'elenco degli OID SNMP per il polling può essere recuperato dallo strumento di monitoraggio SNMP; tuttavia, in questo caso, erano presenti i seguenti fattori preventivi:

- L'amministratore FTD non aveva accesso allo strumento di monitoraggio SNMP
- SNMP versione 3 con autenticazione e crittografia dei dati per la privacy è stato configurato su FTD

Analisi acquisizione

Poiché l'amministratore FTD disponeva delle credenziali per l'autenticazione SNMP versione 3 e la crittografia dei dati, è stato proposto questo piano d'azione:

1. Acquisizione di pacchetti SNMP
2. Salvare le clip e usare le preferenze del protocollo Wireshark SNMP per specificare le

credenziali SNMP versione 3 per decrittografare i pacchetti SNMP versione 3. Le clip decrittografate vengono utilizzate per l'analisi e il recupero degli OID SNMP

Configurare le acquisizioni dei pacchetti SNMP sull'interfaccia utilizzata nella configurazione dell'host snmp-server:

```
<#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	0.325	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_
  
```

Considerazioni principali:

1. Indirizzi/porte di origine e di destinazione SNMP.
2. Impossibile decodificare la PDU del protocollo SNMP. PrivKey è sconosciuto a Wireshark.
3. Il valore della primitiva encryptedPDU.

Azioni consigliate

Le azioni elencate in questa sezione hanno lo scopo di limitare ulteriormente il problema.

Azione 1. Decrittografare le clip SNMP.

Salvare le clip e modificare le preferenze del protocollo SNMP Wireshark per specificare le credenziali SNMP versione 3 per decrittografare i pacchetti.

```
<#root>
```

```
firepower#
```

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

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

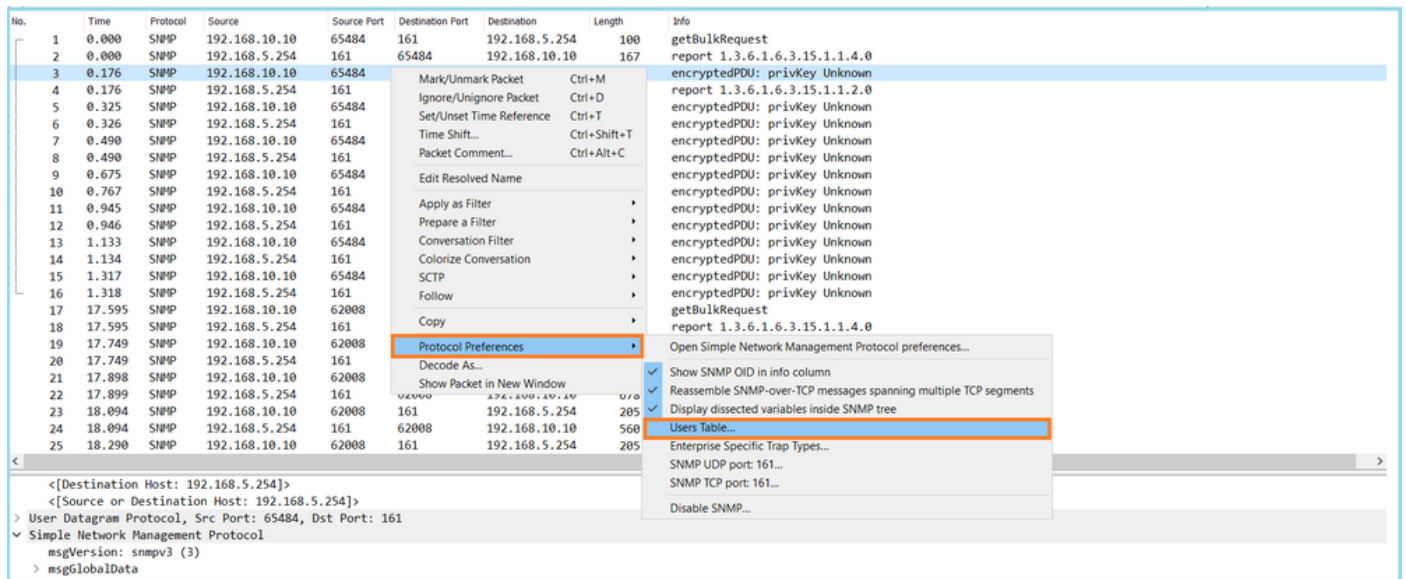
```
Address or name of remote host []? 192.168.10.253
```

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

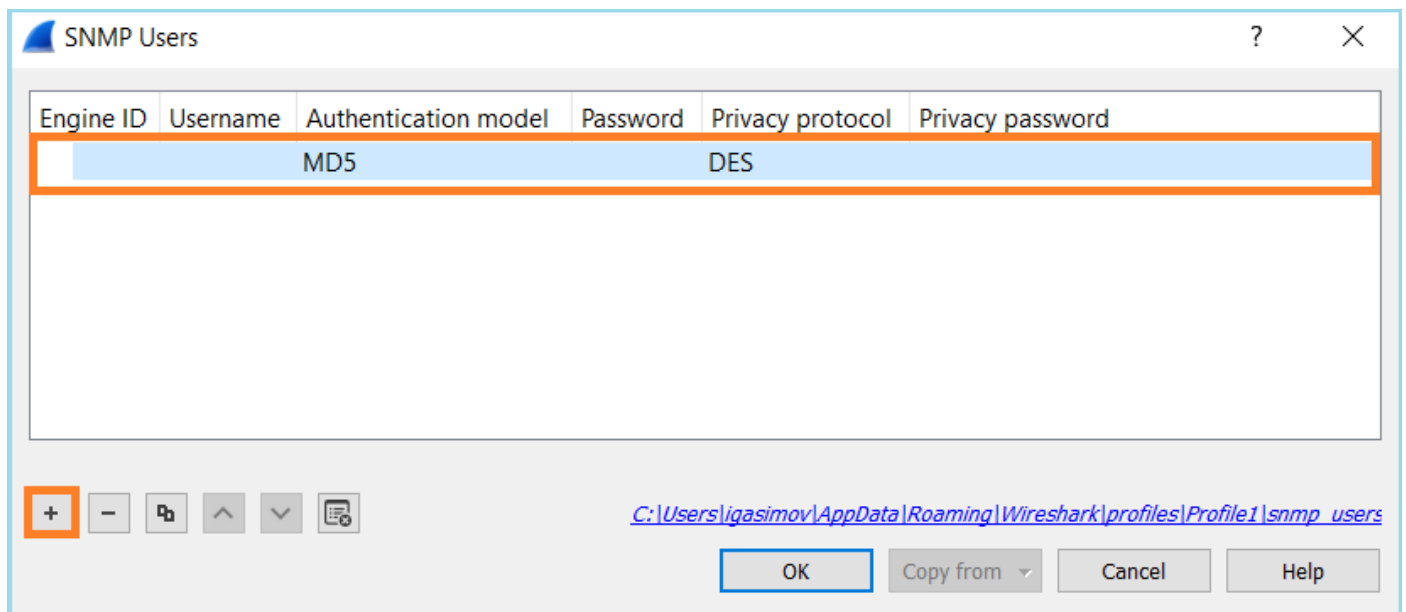
```
!!!!!!
```

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


Aprire il file di acquisizione su Wireshark, selezionare un pacchetto SNMP e selezionare Protocol Preferences > Users Table (Preferenze protocollo > Tabella utenti), come mostrato nell'immagine:



Nella tabella Utenti SNMP sono stati specificati il nome utente SNMP versione 3, il modello di autenticazione, la password di autenticazione, il protocollo di privacy e la password per la privacy (le credenziali effettive non sono mostrate di seguito):



Una volta applicate le impostazioni degli utenti SNMP, Wireshark ha mostrato le PDU SNMP decrittografate:

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 1.3.6.1.4.1.9.9.221.1.1.1.18.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	get-response 1.3.6.1.4.1.9.9.221.1.1.1.19.1.1 1.3.6.1.4.1.9.9.221.1.1.1.19.1.2 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8
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 1.3.6.1.4.1.9.9.392.1.1.1.0 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 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 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 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8


```

< 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)
  
```

Considerazioni principali:

1. Gli strumenti di monitoraggio SNMP hanno utilizzato SNMP getBulkRequest per eseguire query sull'OID 1.3.6.1.4.1.9.9.221.1 padre e sugli OID correlati e per esaminarlo.
2. L'FTD ha risposto a ogni getBulkRequest con get-response contenente OID correlati a 1.3.6.1.4.1.9.9.221.1.

Azione 2. Identificare gli OID SNMP.

[SNMP Object Navigator](#) ha mostrato che OID 1.3.6.1.4.1.9.9.221.1 appartiene al MIB (Management Information Base) denominato CISCO-ENHANCED-MEMPOOL-MIB, come mostrato nell'immagine:

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)
|

Per visualizzare gli OID in formato leggibile in Wireshark:

1. Scaricare il MIB CISCO-ENHANCED-MEMPOOL-MIB e le relative dipendenze, come mostrato nell'immagine:

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 | 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](#)

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. In Wireshark in Modifica > Preferenze > finestra Risoluzione nome, è selezionata l'opzione Abilita risoluzione OID. Nella finestra SMI (percorsi MIB e PIB) specificare la cartella con i MIB scaricati e nei moduli SMI (MIB e PIB). Il CISCO-ENHANCED-MEMPOOL-MIB viene aggiunto automaticamente all'elenco dei moduli:

The screenshot shows the Wireshark interface with the following windows open:

- Wireshark - Preferences:** The 'Name Resolution' section is expanded, and the checkbox for 'Enable OID resolution' is checked.
- SMI Paths:** The 'Directory path' field contains 'C:/Users/Administrator/Downloads/SNMPMIBS'.
- SMI Modules:** The 'Module name' list includes 'CISCO-ENHANCED-MEMPOOL-MIB', which is highlighted.

3. Dopo il riavvio di Wireshark, viene attivata la risoluzione 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::usmStatsNotInTimeInWindows.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::compMemPc


```

✓ 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

```

In base all'output decrittografato del file di acquisizione, lo strumento di monitoraggio SNMP ha eseguito periodicamente (a intervalli di 10 secondi) il polling dei dati sull'utilizzo dei pool di memoria sull'FTD. Come spiegato nell'articolo di TechNote [ASA SNMP Polling for Memory-Related Statistics](#), il polling dell'utilizzo del Global Shared Pool (GSP) con il protocollo SNMP determina un elevato utilizzo della CPU. In questo caso dalle acquisizioni, è chiaro che l'utilizzo del pool condiviso globale è stato periodicamente sottoposto a polling come parte della primitiva getBulkRequest di SNMP.

Per ridurre al minimo i blocchi della CPU causati dal processo SNMP, è stato consigliato di seguire i passaggi di mitigazione per i blocchi della CPU per il protocollo SNMP indicati nell'articolo e di evitare il polling degli OID relativi all'SPG. Senza il sondaggio SNMP per gli OID relativi all'SPG, non sono stati osservati hog della CPU causati dal processo SNMP e la velocità di sovraccarico è diminuita in modo significativo.

Informazioni correlate

- [Guide alla configurazione di Cisco Firepower Management Center](#)
- [Informazioni sulle azioni delle regole delle policy di controllo degli accessi di Firepower Threat Defense](#)
- [Uso di acquisizioni e Packet Tracer di Firepower Threat Defense](#)
- [Informazioni su Wireshark](#)

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).