Analise as capturas de firewall do Firepower para solucionar problemas de rede

Contents

Introdução

Este documento descreve várias técnicas de análise de captura de pacotes que visam solucionar problemas de rede de forma eficaz.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Arquitetura da plataforma Firepower
- Logs de NGFW
- Packet Tracer de NGFW

Além disso, antes de começar a analisar capturas de pacotes, é altamente recomendável atender a estes requisitos:

- Conhecer a operação do protocolo - Não comece a verificar uma captura de pacote se não entender como o protocolo capturado opera.
- Conhecer a topologia - Você deve conhecer os dispositivos de trânsito de ponta a ponta. Se isso não for possível, você deve pelo menos conhecer os dispositivos upstream e downstream.
- Conheça o dispositivo - Você deve saber como o dispositivo lida com pacotes, quais são as interfaces envolvidas (entrada/saída), qual é a arquitetura do dispositivo e quais são os vários pontos de captura.
- Conhecer a configuração - Você deve saber como um fluxo de pacote deve ser tratado pelo dispositivo em termos de:

    - Interface de roteamento/saída
    - Políticas aplicadas
    - Tradução de Endereço de Rede (NAT)
- Conhecer as ferramentas disponíveis - Junto com as capturas, é recomendável estar pronto para aplicar outras ferramentas e técnicas (como registro e rastreadores) e, se necessário, correlacioná-las com os pacotes capturados.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- A maioria dos cenários é baseada no FP4140 executando o software FTD 6.5.x.
- FMC executando o software 6.5.x.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

A captura de pacotes é uma das ferramentas de solução de problemas mais negligenciadas disponíveis atualmente. Diariamente, o Cisco TAC resolve muitos problemas com a análise dos dados capturados.

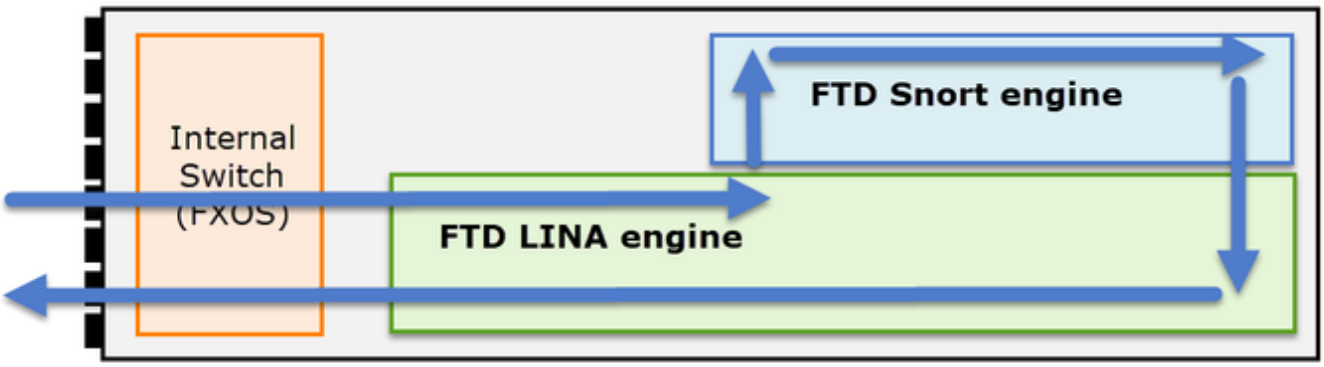
O objetivo deste documento é ajudar os engenheiros de rede e segurança a identificar e solucionar problemas comuns de rede com base principalmente na análise de captura de pacotes.

Todos os cenários apresentados neste documento são baseados em casos de usuários reais vistos no Centro de Assistência Técnica da Cisco (TAC).

O documento aborda as capturas de pacotes do ponto de vista do Cisco Next-Generation Firewall (NGFW), mas os mesmos conceitos também se aplicam a outros tipos de dispositivos.

Como coletar e exportar capturas da família de produtos NGFW?

No caso de um dispositivo Firepower (1xxx, 21xx, 41xx, 93xx) e um aplicativo Firepower Threat Defense (FTD), um processamento de pacote pode ser visualizado conforme mostrado na imagem.



1. Um pacote entra na interface de entrada e é tratado pelo switch interno do chassi.
2. O pacote entra no mecanismo FTD Lina, que faz principalmente verificações de L3/L4.
3. Se a política exigir que o pacote seja inspecionado pelo mecanismo Snort (principalmente inspeção L7).
4. O mecanismo Snort retorna um veredito para o pacote.
5. O mecanismo LINA descarta ou encaminha o pacote de acordo com a conclusão do Snort.
6. O pacote sai do chassi através do switch interno do chassi.

Com base na arquitetura mostrada, as capturas de FTD podem ser realizadas em três (3) locais diferentes:

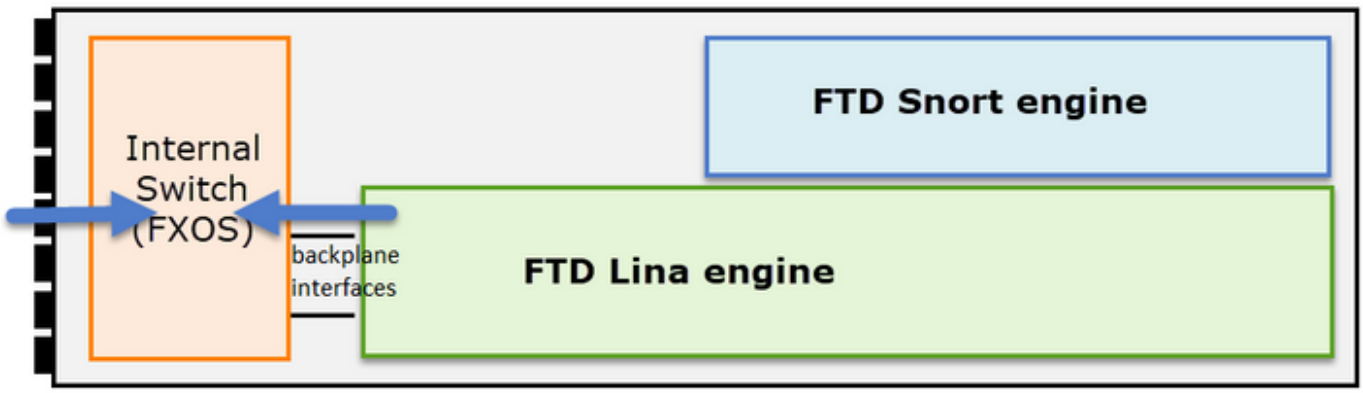
- FXOS
- Mecanismo FTD Lina
- Mecanismo Snort de FTD

Coletar Capturas FXOS

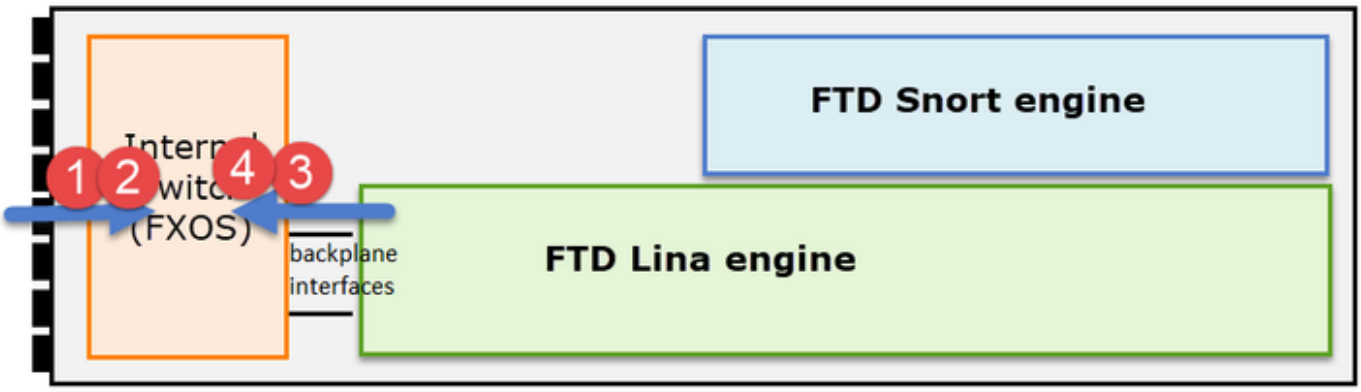
O processo é descrito neste documento:

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

As capturas de FXOS só podem ser feitas na direção de entrada do ponto de vista interno do switch são mostradas na imagem aqui.



Aqui são mostrados dois pontos de captura por direção (devido à arquitetura de switch interno).



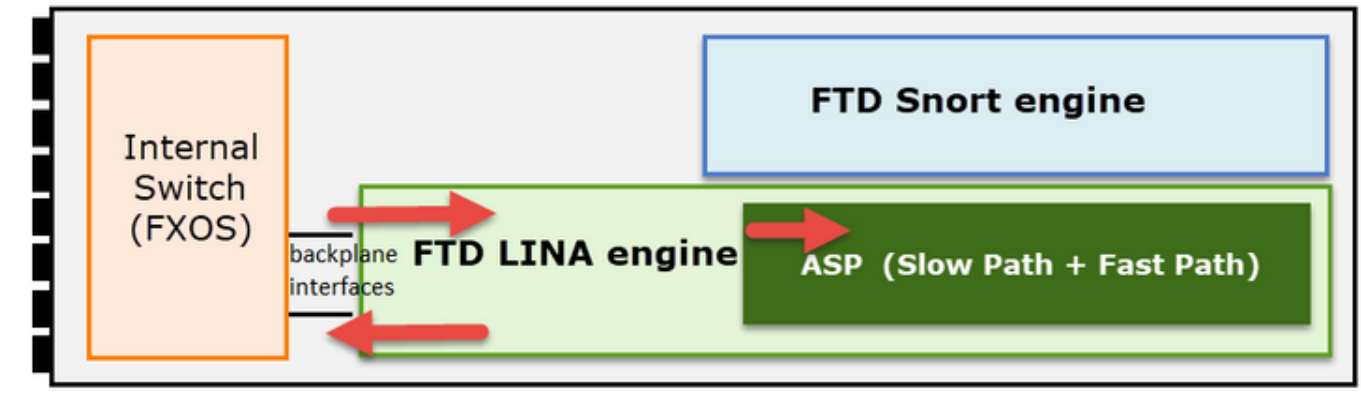
Os pacotes capturados nos pontos 2, 3 e 4 têm uma tag de rede virtual (VNTag).

Observação: as capturas no nível do chassi FXOS estão disponíveis apenas nas plataformas FP41xx e FP93xx. FP1xxx e FP21xx não fornecem esse recurso.

Habilitar e Coletar Capturas Lina do FTD

Principais pontos de captura:

- Interface de entrada
- Interface de saída
- Caminho de segurança acelerado (ASP)



Você pode usar a interface do usuário do Firepower Management Center (FMC UI) ou a CLI do FTD para ativar e coletar as capturas do FTD Lina.

Habilite a captura a partir do CLI na interface INSIDE:

<#root>

```
firepower#
```

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

Essa captura corresponde ao tráfego entre os IPs 192.168.103.1 e 192.168.101.1 em ambas as direções.

Habilite a captura ASP para ver todos os pacotes descartados pelo mecanismo FTD Lina:

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

Exportar uma captura Lina de FTD para um servidor FTP:

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

Exportar uma captura FTD Lina para um servidor TFTP:

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

A partir da versão 6.2.x do FMC, você pode ativar e coletar capturas FTD Lina da interface do FMC.

Outra maneira de coletar capturas de FTD de um firewall gerenciado pelo FMC é essa.

Passo 1

No caso de captura LINA ou ASP, copie a captura para o disco FTD.

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

Source capture name [capin]?

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

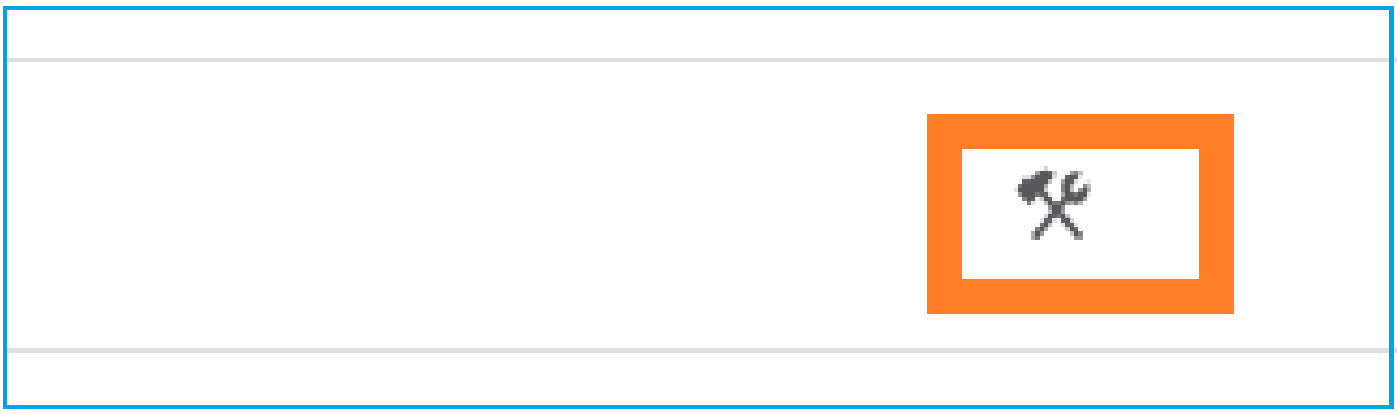
Passo 2

Navegue até o modo especialista, localize a captura salva e copie-a para o local /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
```

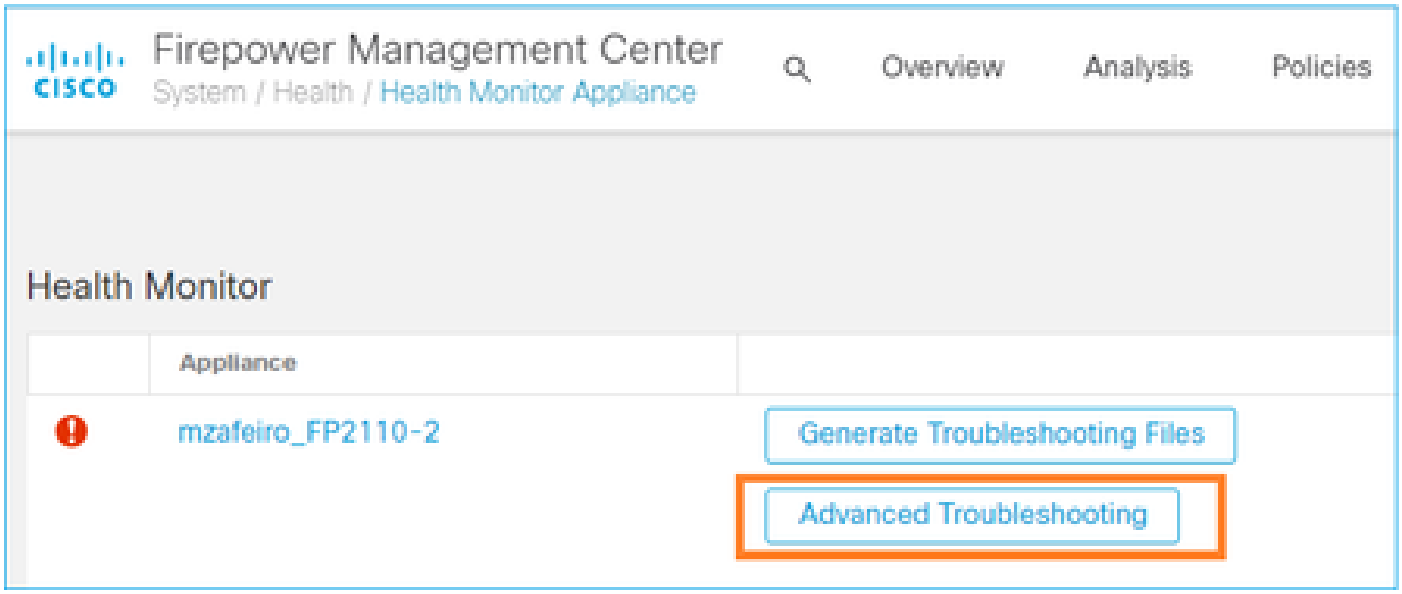
Etapa 3

Faça login no FMC que gerencia o FTD e navegue até Devices > Device Management. Localize o dispositivo FTD e selecione o ícone Solução de problemas:

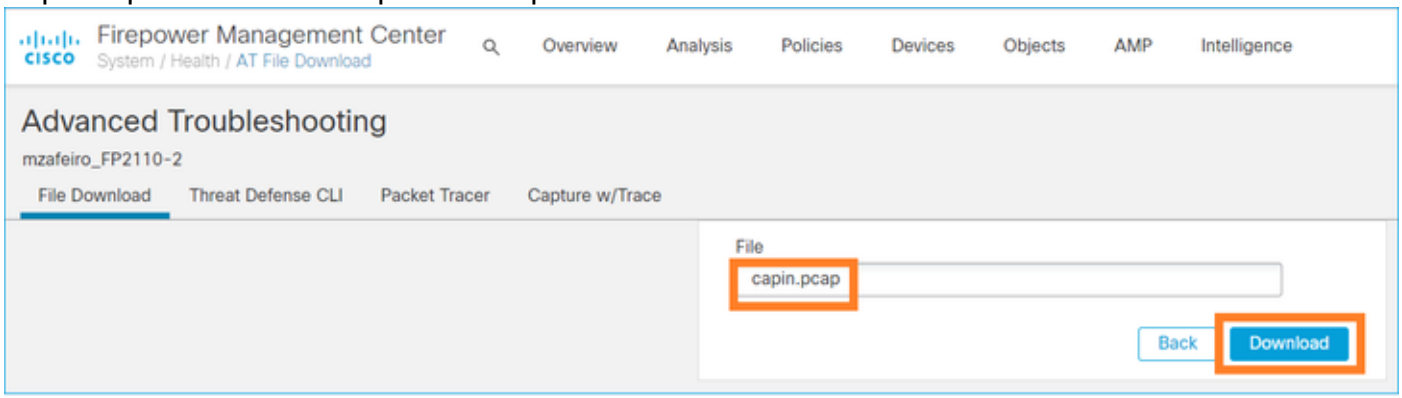


Passo 4

Selecione Solução de problemas avançada:



Especifique o nome do arquivo de captura e selecione Download:

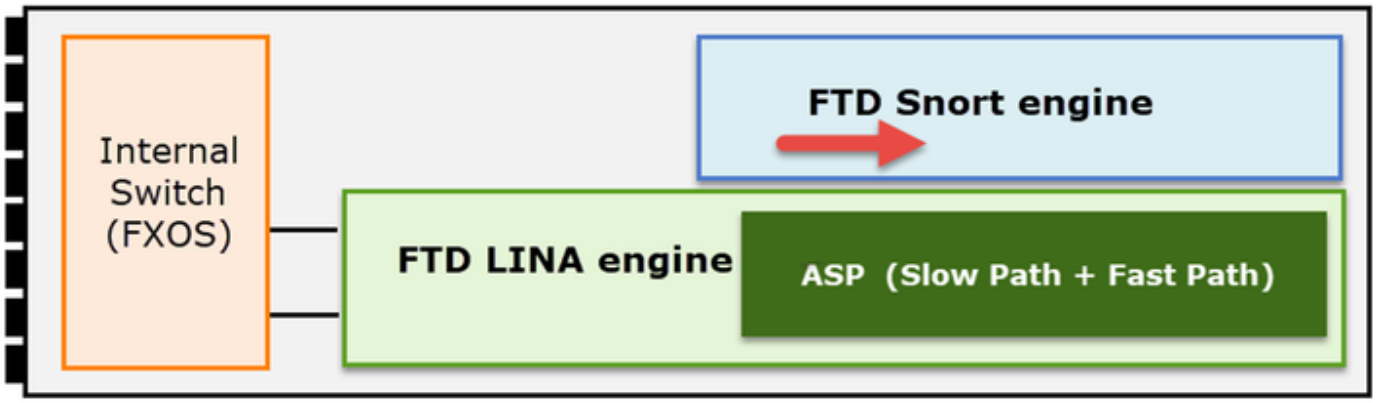


Para obter mais exemplos sobre como habilitar/coletar capturas da interface do usuário do FMC, consulte este documento:

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

Habilitar e Coletar Capturas Snort de FTD

O ponto de captura é mostrado na imagem aqui.



Habilitar captura no nível do Snort:

```
<#root>

>

capture-traffic


Please choose domain to capture traffic from:
  0 - br1
  1 - Router

Selection?

1


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

-n host 192.168.101.1
```

Para gravar a captura em um arquivo com o nome capture.pcap e copiá-lo via FTP para um servidor remoto:

```
<#root>

>

capture-traffic


Please choose domain to capture traffic from:
  0 - br1
  1 - Router

Selection?

1


Please specify tcpdump options desired.
```

```
(or enter '?' for a list of supported options)
Options:
```

-w capture.pcap host 192.168.101.1

CTRL + C <- to stop the capture

```
>
```

file copy 10.229.22.136 ftp / capture.pcap

```
Enter password for ftp@10.229.22.136:
Copying capture.pcap
Copy successful.

>
```

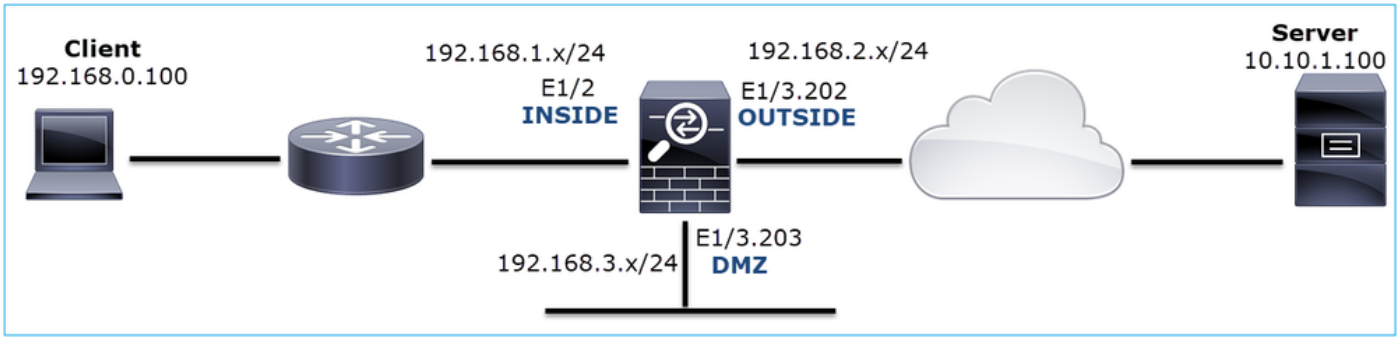
Para obter mais exemplos de captura em nível de Snort que incluam diferentes filtros de captura, marque este documento:

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

Troubleshooting

Caso 1. Sem TCP SYN na interface de saída

A topologia é mostrada na imagem aqui:



Descrição do problema: o HTTP não funciona

Fluxo afetado:

IP orig.: 192.168.0.100

IP do Horário de Verão: 10.10.1.100

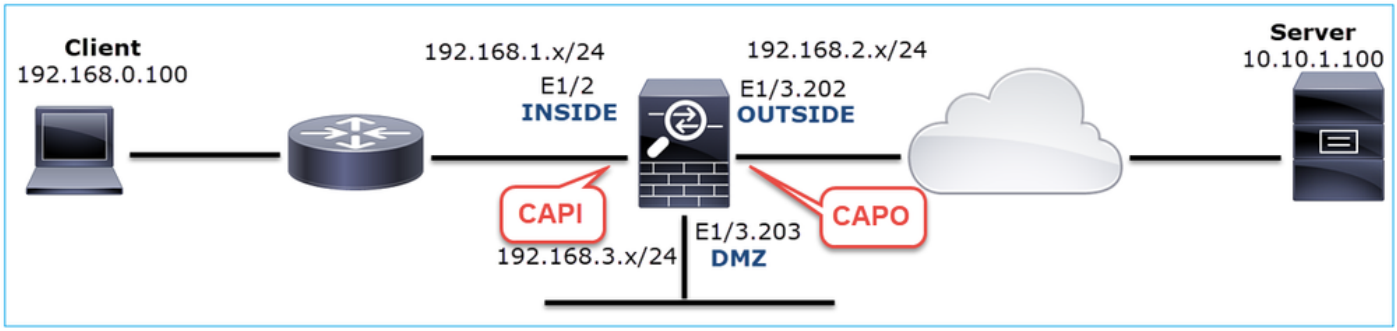
Protocolo: TCP 80

Capturar análise

Habilitar capturas no mecanismo LINA do FTD:

<#root>

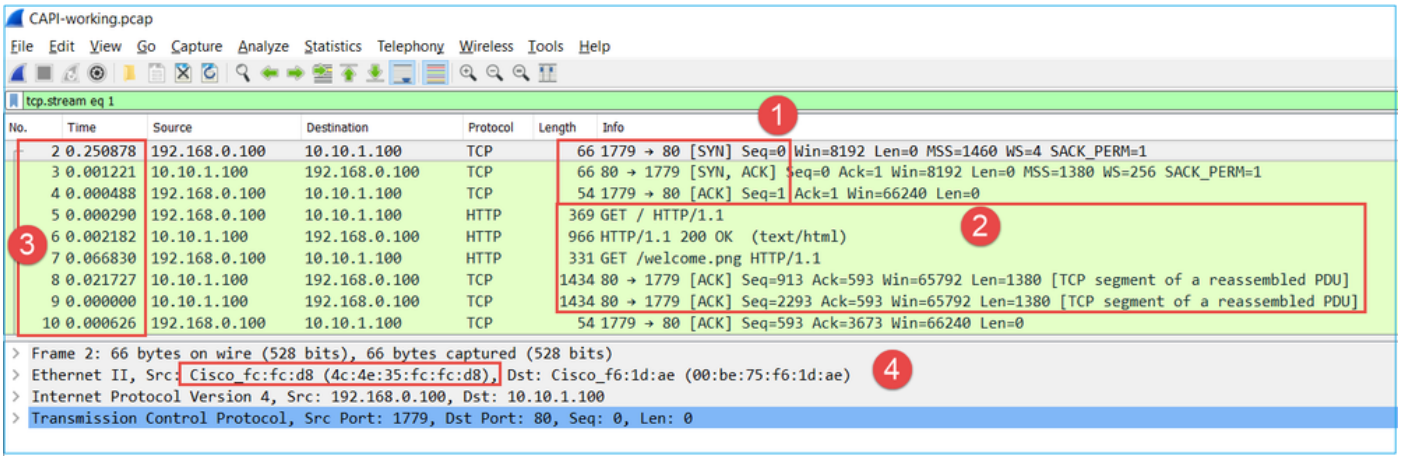
```
firepower#
capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100
firepower#
capture CAPO int OUTSIDE match ip host 192.168.0.100 host 10.10.1.100
```



Capturas - Cenário Funcional:

Como linha de base, é sempre muito útil ter capturas de um cenário funcional.

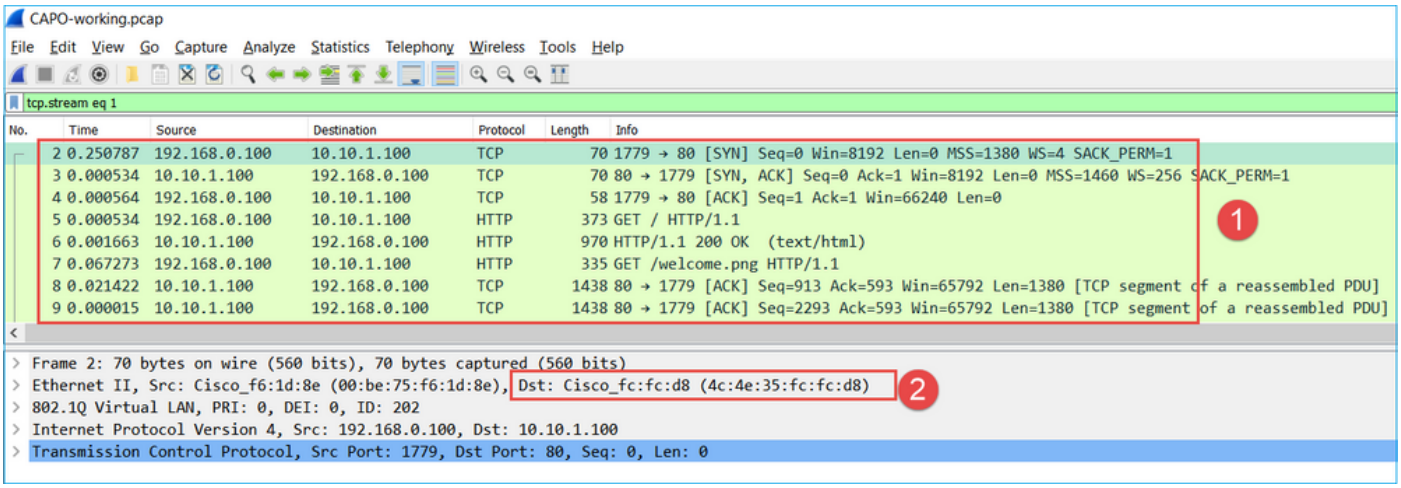
A captura feita na interface NGFW INSIDE é como mostrado na imagem:



Pontos principais:

1. Handshake triplo do TCP.
2. Intercâmbio de dados bidirecional.
3. Não há atrasos entre os pacotes (com base na diferença de tempo entre os pacotes).
4. O MAC origem é o dispositivo downstream correto.

A captura feita na interface NGFW OUTSIDE é mostrada na imagem aqui:



Pontos principais:

1. Os mesmos dados da captura CAPI.
2. O MAC destino é o dispositivo upstream correto.

Capturas - cenário não funcional

Na CLI do dispositivo, as capturas são assim:

<#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
```

Conteúdo CAPI:

<#root>

```
firepower#
```

show capture CAPI

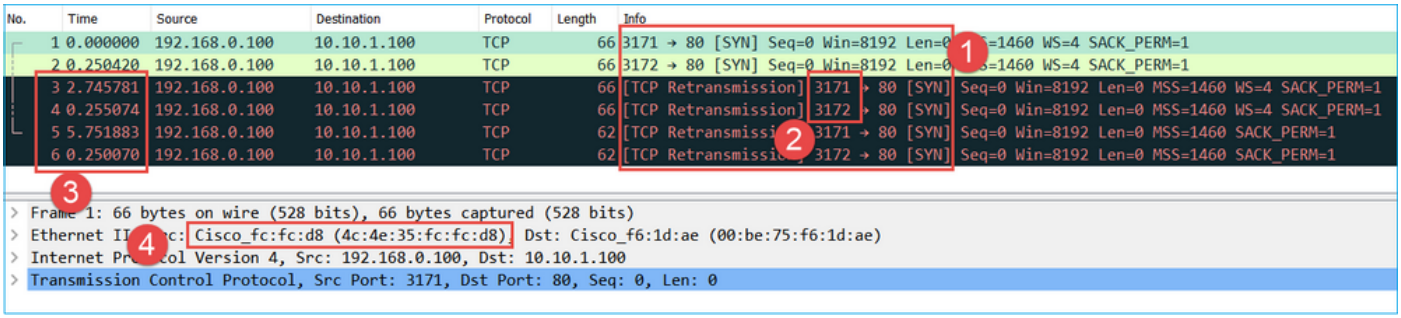
```
6 packets captured

   1: 11:47:46.911482    192.168.0.100.3171 > 10.10.1.100.80:
```

```
S

 1089825363:1089825363(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
   2: 11:47:47.161902   192.168.0.100.3172 > 10.10.1.100.80:

S

 3981048763:3981048763(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
   3: 11:47:49.907683   192.168.0.100.3171 > 10.10.1.100.80:

S

 1089825363:1089825363(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
   4: 11:47:50.162757   192.168.0.100.3172 > 10.10.1.100.80:

S

 3981048763:3981048763(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
   5: 11:47:55.914640   192.168.0.100.3171 > 10.10.1.100.80:

S

 1089825363:1089825363(0) win 8192 <mss 1460,nop,nop,sackOK>
   6: 11:47:56.164710   192.168.0.100.3172 > 10.10.1.100.80:

S

 3981048763:3981048763(0) win 8192 <mss 1460,nop,nop,sackOK>
```

<#root>

```
firepower#

show capture CAPO


0 packet captured


0 packet shown
```

Esta é a imagem da captura CAPI no Wireshark:



Pontos principais:

1. Somente os pacotes TCP SYN são vistos (sem handshake triplo do TCP).
2. Há duas sessões TCP (portas de origem 3171 e 3172) que não podem ser estabelecidas. O

cliente de origem reenvia os pacotes TCP SYN. Esses pacotes retransmitidos são identificados pelo Wireshark como retransmissões de TCP.
3. As retransmissões de TCP ocorrem a cada ~3 e depois a cada 6 segundos etc.
4. O endereço MAC origem é proveniente do dispositivo downstream correto.

Com base nas duas capturas, pode concluir-se que:

- Um pacote de 5 tuplas específicas (src/dst IP, src/dst port, protocol) chega ao firewall na interface esperada (INSIDE).
- Um pacote não deixa o firewall na interface esperada (EXTERNA).

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Verifique o rastreamento de um pacote emulado.

Use a ferramenta packet-tracer para ver como um pacote deve ser tratado pelo firewall. Caso o pacote seja descartado pela política de acesso do firewall, o rastreamento do pacote emulado será semelhante a esta saída:

<#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
```

Ação 2. Verifique os rastreamentos de pacotes ativos.

Ative o rastreamento de pacotes para verificar como os pacotes TCP SYN reais são tratados pelo firewall. Por padrão, somente os primeiros 50 pacotes de entrada são rastreados:

```
<#root>

firepower#

capture CAPI trace
```

Limpe o buffer de captura:

```
<#root>

firepower#

clear capture /all
```

Caso o pacote seja descartado pela Política de Acesso do firewall, o rastreamento será semelhante a esta saída:

```
<#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
```

Ação 3. Verifique os logs do FTD Lina.

Para configurar o Syslog no FTD via FMC, consulte este documento:

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

É altamente recomendável ter um servidor Syslog externo configurado para logs FTD Lina. Se não houver um servidor Syslog remoto configurado, habilite os logs de buffer local no firewall enquanto soluciona os problemas. A configuração de log mostrada neste exemplo é um bom ponto inicial:

```
<#root>

firepower#

show run logging

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

Defina o pager do terminal como 24 linhas para controlar o pager do terminal:

```
<#root>

firepower#

terminal pager 24
```

Limpe o buffer de captura:

```
<#root>

firepower#

clear logging buffer
```

Teste a conexão e verifique os logs com um filtro do analisador. Neste exemplo, os pacotes são descartados pela Política de acesso de 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
```

Ação 4. Verifique as quedas de firewall ASP.

Se você suspeitar que o pacote foi descartado pelo firewall, poderá ver os contadores de todos os pacotes descartados pelo firewall no nível do 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
```

Você pode habilitar as capturas para ver todas as quedas de nível de software do ASP:

<#root>

```
firepower#
```

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

Dica: se você não estiver interessado no conteúdo do pacote, poderá capturar apenas os cabeçalhos do pacote (opção apenas cabeçalhos). Isso permite capturar muito mais pacotes no buffer de captura. Além disso, você pode aumentar o tamanho do buffer de captura (por padrão é 500Kbytes) para um valor de até 32 Mbytes (opção de buffer). Finalmente, a partir do FTD versão 6.3, a opção de tamanho de arquivo permite configurar um arquivo de captura de até 10 GBytes. Nesse caso, você só pode ver o conteúdo da captura em um formato pcap.

Para verificar o conteúdo da captura, você pode usar um filtro para restringir sua pesquisa:

<#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
```

Nesse caso, como os pacotes já estão rastreados no nível da interface, o motivo para a queda não é mencionado na captura ASP. Lembre-se de que um pacote só pode ser rastreado em um lugar (interface de entrada ou queda de ASP). Nesse caso, é recomendável usar vários descartes de ASP e definir um motivo específico para o descarte. Aqui está uma abordagem recomendada:

1. Limpe os contadores de queda ASP atuais:

```
<#root>

firepower#

clear asp drop
```

2. Envie o fluxo cujos problemas você soluciona através do firewall (execute um teste).

3. Verifique novamente os contadores suspensos do ASP e anote os que foram aumentados.

```
<#root>

firepower#

show asp drop

Frame drop:
  No route to host (

no-route

)                                               234
  Flow is denied by configured rule (

acl-drop

)                                  71
```

4. Habilite a(s) captura(ões) ASP para as quedas específicas vistas:

```
<#root>

firepower#
```

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

firepower#

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

5. Envie o fluxo cujos problemas você soluciona através do firewall (execute um teste).

6. Verifique as capturas ASP. Nesse caso, os pacotes foram descartados devido a uma rota ausente:

```
<#root>

firepower#

show capture ASP_NO_ROUTE | include 192.168.0.100.*10.10.1.100

  93: 07:53:52.381663   192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss
  95: 07:53:52.632337   192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss
 101: 07:53:55.375392   192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss
 102: 07:53:55.626386   192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss
 116: 07:54:01.376231   192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss
 117: 07:54:01.626310   192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss
```

Ação 5. Verifique a tabela de conexão FTD Lina.

Pode haver casos em que você espera que o pacote saia da interface 'X', mas por quaisquer razões ele sai da interface 'Y'. A determinação da interface de saída do firewall é baseada nesta ordem de operação:

1. Pesquisa de Conexão Estabelecida
2. Consulta de conversão de endereço de rede (NAT) - A fase UN-NAT (NAT de destino) tem precedência sobre a pesquisa de PBR e de rota.
3. Roteamento baseado em políticas (PBR)
4. Pesquisa na tabela de roteamento

Para verificar a tabela de conexão do 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

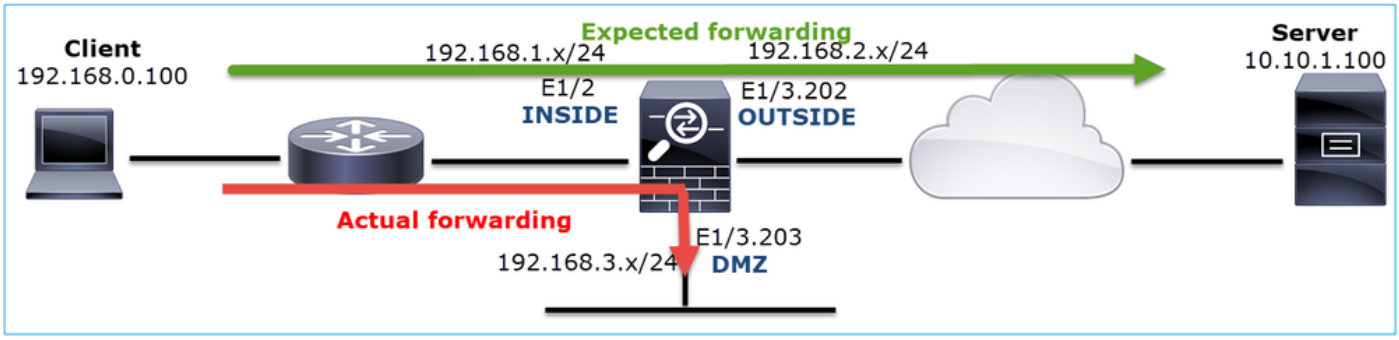
, idle 0:00:01, bytes 0, flags

aA N1

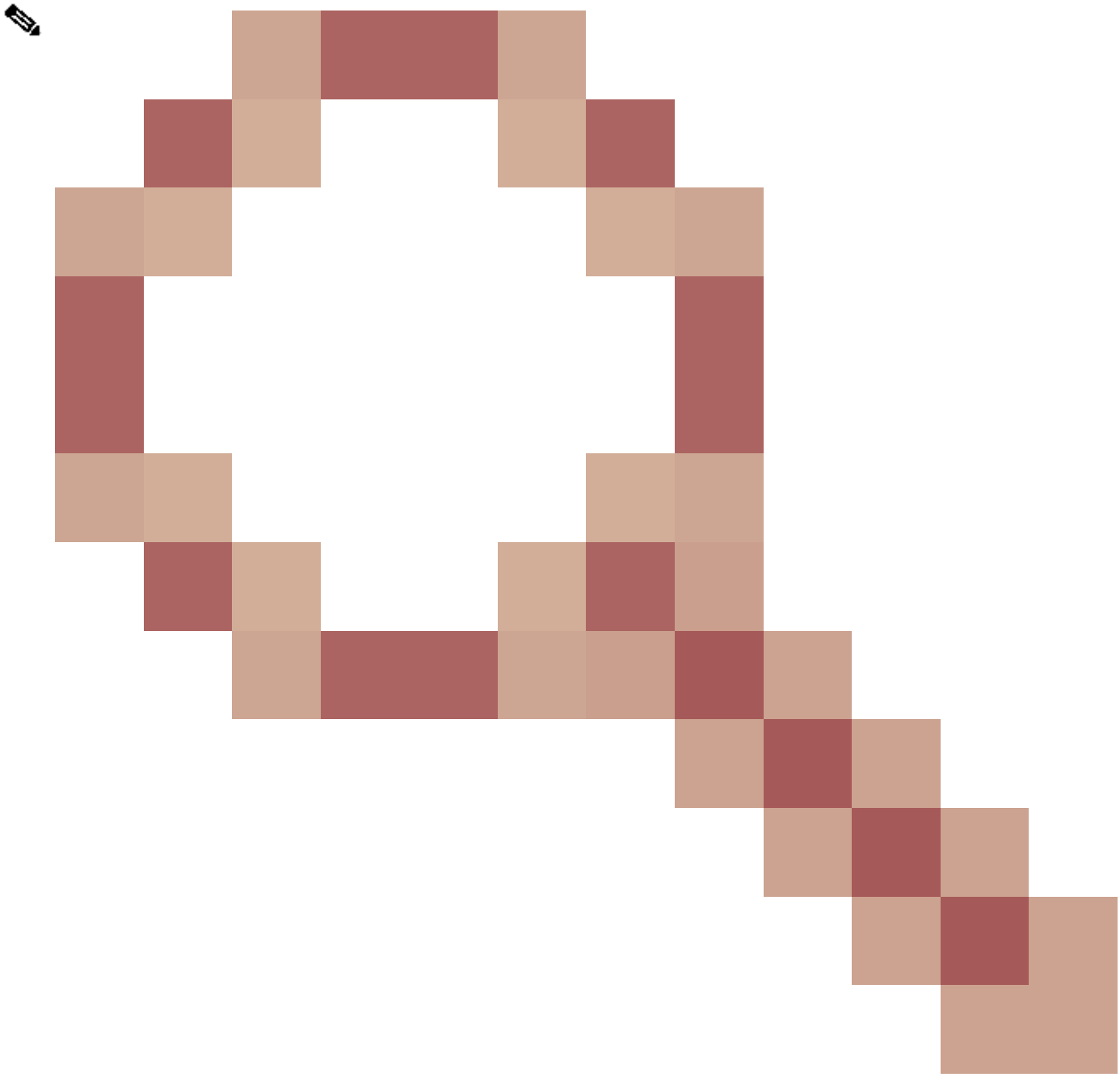
Pontos principais:

- Com base nas flags (Aa), a conexão é embrionária (semiaberta - somente TCP SYN foi visto pelo firewall).
- Com base nas portas de origem/destino, a interface de entrada é INSIDE e a interface de saída é DMZ.

Isso pode ser visualizado na imagem aqui:



Observação: como todas as interfaces FTD têm um nível de segurança 0, a ordem da interface na saída show conn é baseada no número da interface. Especificamente, a interface com número vpif-num mais alto (número de interface da plataforma virtual) é selecionada como interna, enquanto a interface com número vpif-inferior é selecionada como externa. Você pode ver o valor de interface vpif com o comando show interface detail. Aprimoramento relacionado, o bug da Cisco ID CSCvi15290



ENH: FTD mostra a direcionalidade da conexão na saída 'show conn' do 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
```

Observação: a partir da versão 6.5 do software Firepower, a versão 9.13.x do ASA, as saídas dos comandos show conn long e show conn detail fornecem informações sobre o iniciador e o respondedor da conexão

Saída 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
```

Saída 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
```

Além disso, o comando show conn long exibe os IPs com NAT dentro de um parêntese no caso de uma conversão de endereço de rede:

```
<#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
```

Ação 6. Verifique o cache do firewall Address Resolution Protocol (ARP).

Se o firewall não puder resolver o próximo salto, o firewall descarta silenciosamente o pacote original (TCP SYN nesse caso) e envia continuamente Solicitações ARP até resolver o próximo salto.

Para ver o cache ARP do firewall, use o comando:

<#root>

```
firepower#
```

show arp

Além disso, para verificar se há hosts não resolvidos, você pode usar o 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

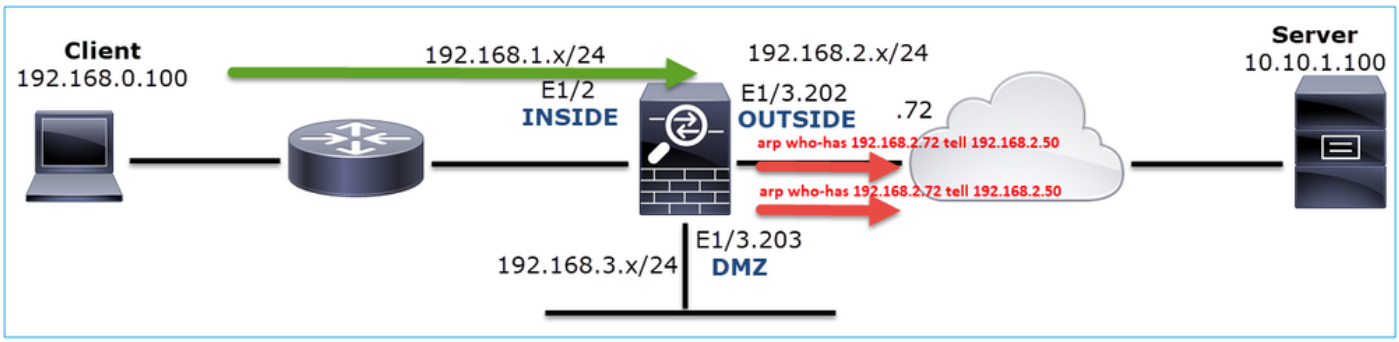
```
        Maximum Unresolved hosts: 2
```

Se quiser verificar mais a operação ARP, você pode ativar uma captura específica do 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
```

Nesta saída, o firewall (192.168.2.50) tenta resolver o próximo salto (192.168.2.72), mas não há resposta ARP



A saída aqui mostra um cenário funcional com resolução ARP apropriada:

<#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
```

Caso não haja uma entrada ARP no local, um rastreamento de um pacote TCP SYN ao vivo 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

Como pode ser visto na saída, o trace mostra Action: allow mesmo quando o próximo salto não está acessível e o pacote é descartado silenciosamente pelo firewall! Nesse caso, a ferramenta packet-tracer também deve ser verificada, pois fornece uma saída mais precisa:

<#root>

```
firepower#
```

packet-tracer input INSIDE tcp 192.168.0.100 1111 10.10.1.100 80

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

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
MAC Access list

Phase: 3
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 192.168.2.72 using egress ifc  OUTSIDE
…

Phase: 14
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 4816, packet dispatched to next module
…
Phase: 17
Type: ROUTE-LOOKUP
```

```
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 192.168.2.72 using egress ifc  OUTSIDE

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

Drop-reason: (no-v4-adjacency) No valid V4 adjacency, Drop-location: frame 0x00005647a4e86109 flow (NA)/

Em versões recentes do ASA/Firepower, a mensagem anterior foi otimizada para:

<#root>

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

Check ARP table (show arp) has entry for nexthop

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

Causas possíveis e resumo das ações recomendadas

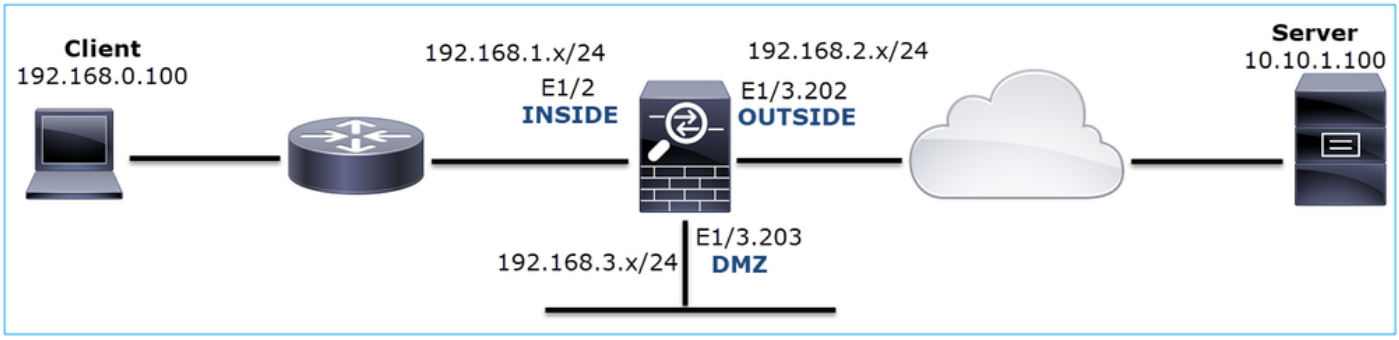
Se você vir apenas um pacote TCP SYN nas interfaces de entrada, mas nenhum pacote TCP SYN enviado para fora da interface de saída esperada, algumas causas possíveis são:

Possível causa	Ações recomendadas
O pacote é descartado pela política de acesso de firewall.	• Use packet-tracer ou capture w/trace para ver como o firewall lida com o pacote. • Verifique os logs do firewall. • Verifique as quedas de firewall ASP (show asp drop ou capture type asp-drop). • Verifique os eventos de conexão do FMC. Isso pressupõe que a regra tenha o registro em log habilitado.
O filtro de captura está errado.	• Use packet-tracer ou capture w/trace para ver se há conversão de NAT que modifique o IP de origem ou destino. Nesse caso, ajuste o filtro de captura.

	• A saída do comando show conn long mostra os IPs com NAT.
O pacote é enviado para uma interface de saída diferente.	• Use packet-tracer ou capture w/trace para ver como o firewall lida com o pacote. Lembre-se da ordem das operações que consideram a determinação da interface de saída, a conexão atual, UN-NAT, PBR e a consulta da tabela de roteamento. • Verifique os logs do firewall. • Verifique a tabela de conexão do firewall (show conn). Se o pacote for enviado a uma interface errada porque corresponde a uma conexão atual, use o comando clear conn address e especifique a tupla 5 da conexão que você deseja limpar.
Não há rota em direção ao destino.	• Use packet-tracer ou capture w/trace para ver como o firewall lida com o pacote. • Verifique as quedas de firewall ASP (show asp drop) para saber o motivo da queda de no-route.
Não há entrada ARP na interface de saída.	• Verifique o cache ARP do firewall (show arp). • Use o packet-tracer para ver se há uma adjacência válida.
A interface de saída está inoperante.	Verifique a saída do comando show interface ip brief no firewall e verifique o status da interface.

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

Esta imagem mostra a topologia:



Descrição do problema: o HTTP não funciona

Fluxo afetado:

IP orig.: 192.168.0.100

IP do Horário de Verão: 10.10.1.100

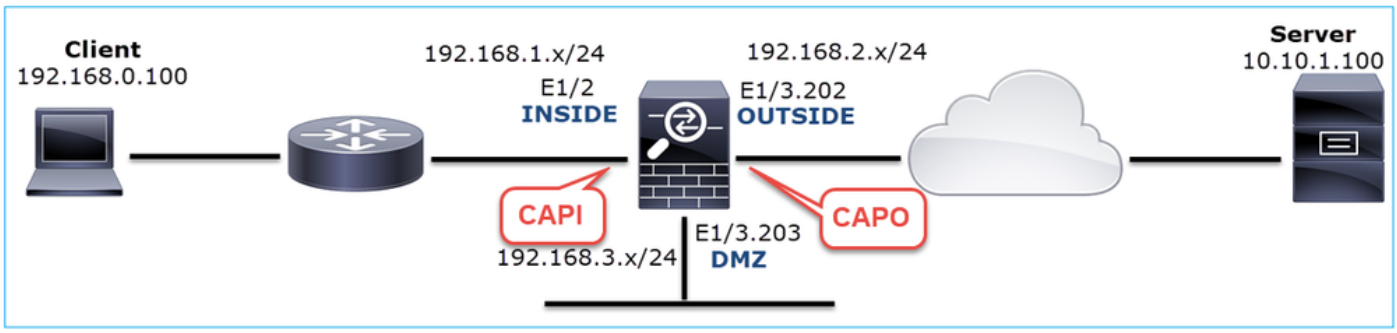
Protocolo: TCP 80

Capturar análise

Habilitar capturas no mecanismo LINA do FTD.

<#root>

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



Capturas - cenário não funcional:

Esta é a aparência das capturas na CLI do dispositivo:

<#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
```

Conteúdo CAPI:

<#root>

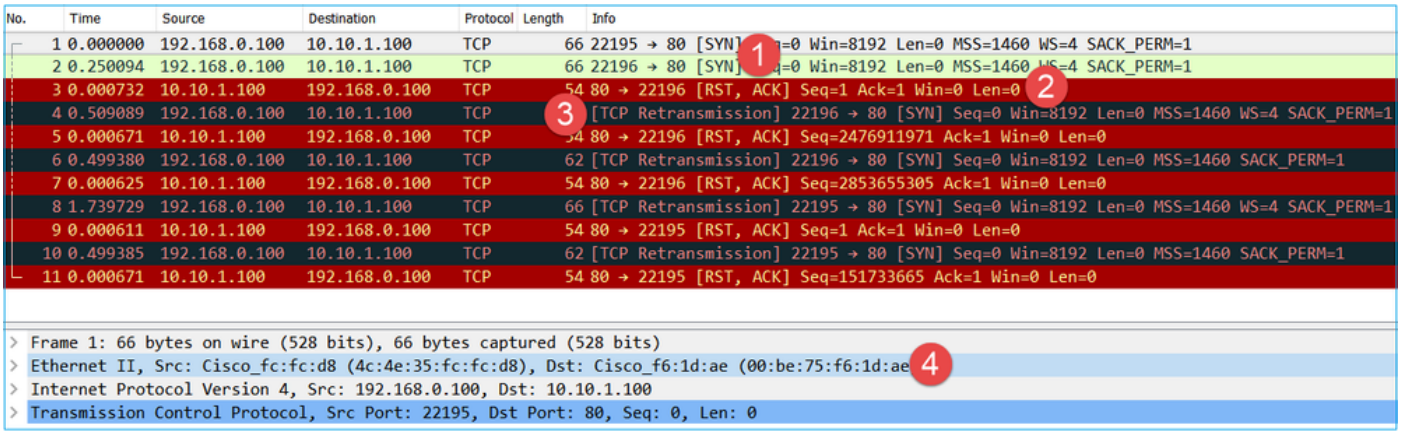
```
firepower#

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>
...
```

Conteúdo do 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>
```

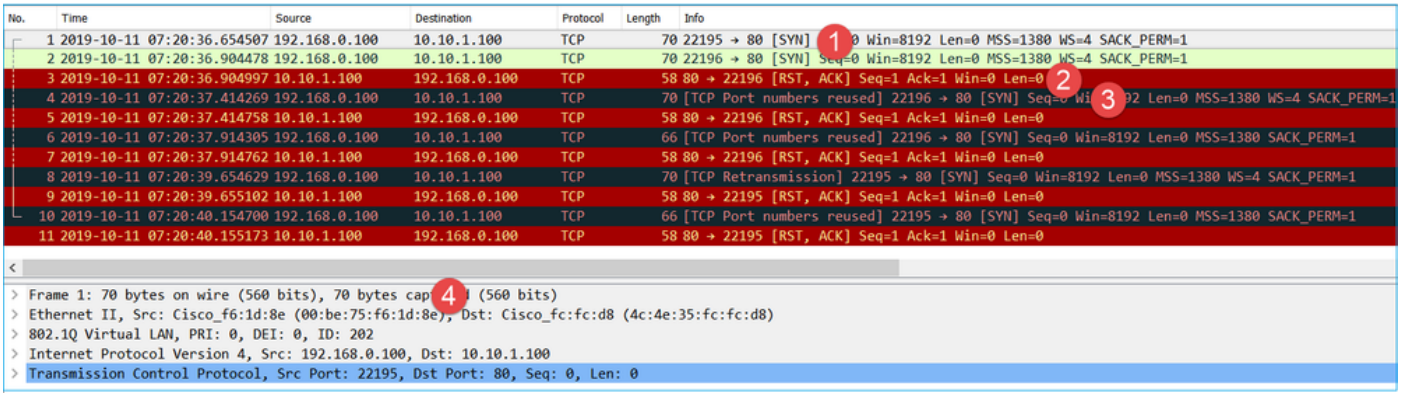
Esta imagem mostra a captura de CAPI no Wireshark.



Pontos principais:

1. A origem envia um pacote TCP SYN.
2. Um TCP RST é enviado para a origem.
3. A origem retransmite os pacotes TCP SYN.
4. Os endereços MAC estão corretos (nos pacotes de entrada, o endereço MAC origem pertence ao roteador downstream, o endereço MAC destino pertence à interface INTERNA do firewall).

Esta imagem mostra a captura de CAPO no Wireshark:



Pontos principais:

1. A origem envia um pacote TCP SYN.
2. Um TCP RST chega à interface EXTERNA.
3. A origem retransmite os pacotes TCP SYN.
4. Os endereços MAC estão corretos (nos pacotes de saída, o firewall OUTSIDE é o MAC origem, o roteador upstream é o MAC destino).

Com base nas duas capturas, pode concluir-se que:

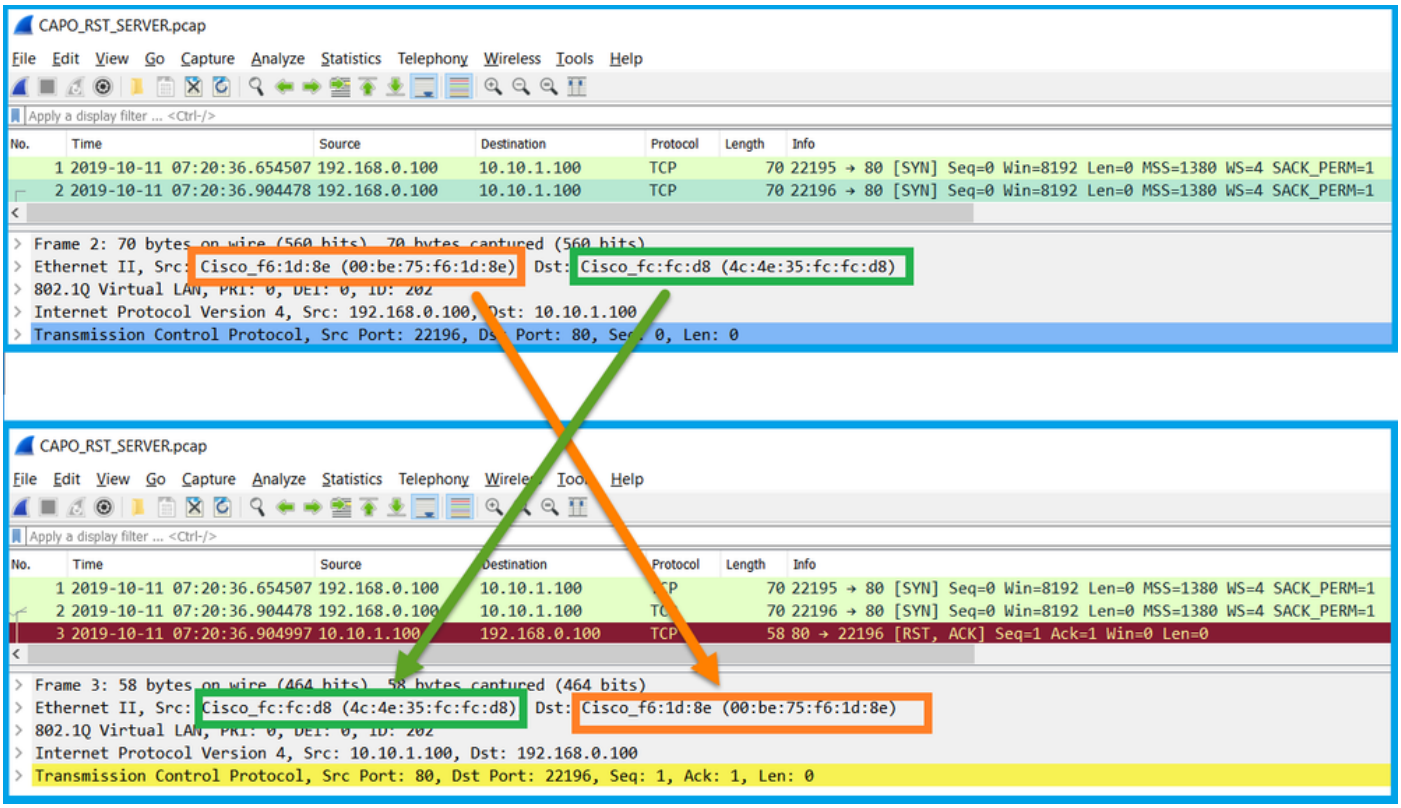
- O handshake triplo do TCP entre o cliente e o servidor não é concluído
- Há um TCP RST que chega à interface de saída do firewall
- O firewall "fala" com os dispositivos upstream e downstream apropriados (com base nos endereços MAC)

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Verifique o endereço MAC origem que envia o TCP RST.

Verifique se o MAC de destino visto no pacote TCP SYN é o mesmo que o MAC de origem visto no pacote TCP RST.

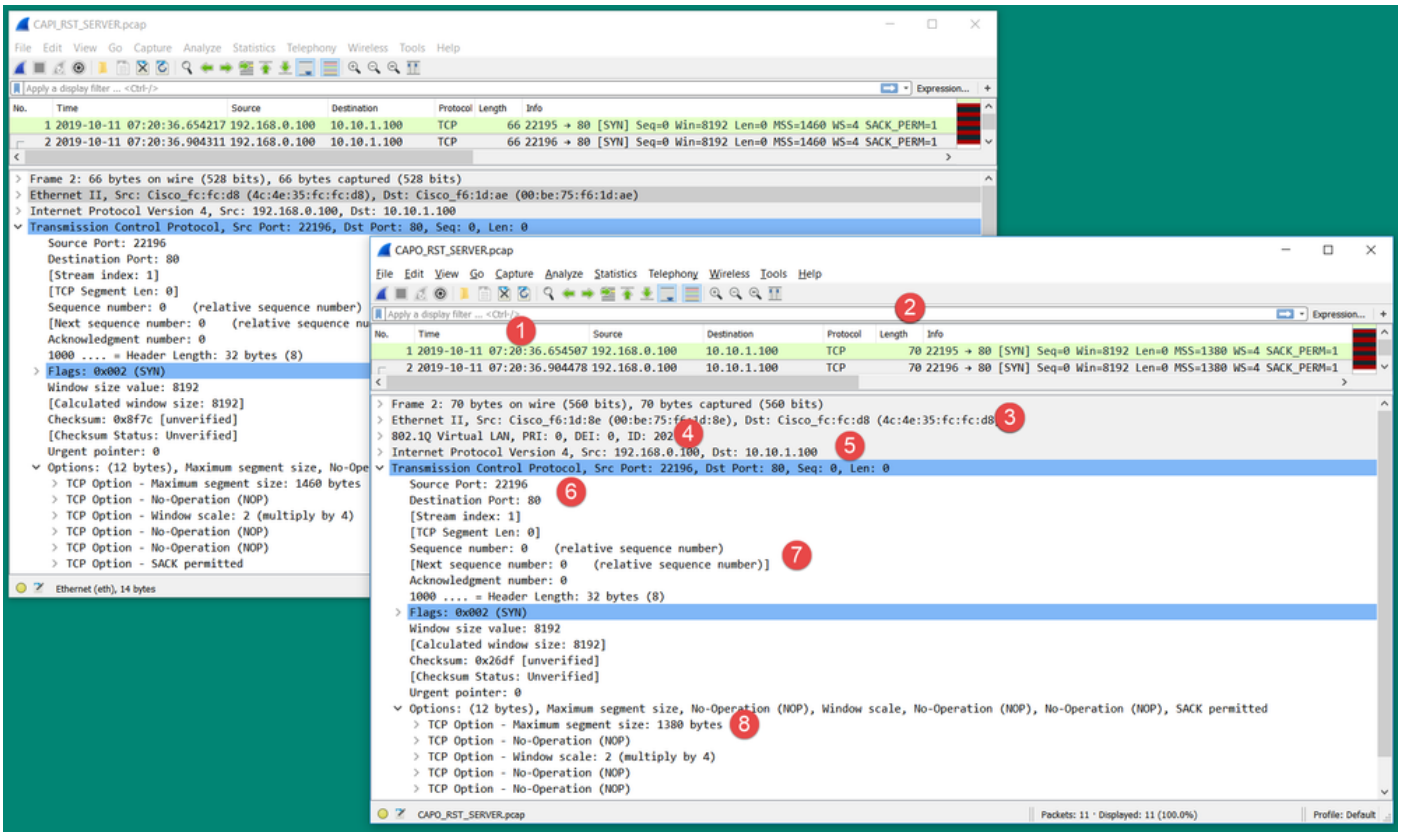


Esta verificação tem como objetivo confirmar duas coisas:

- Verifique se não há fluxo assimétrico.
- Verifique se o MAC pertence ao dispositivo upstream esperado.

Ação 2. Comparar pacotes de entrada e saída.

Compare visualmente os 2 pacotes no Wireshark para verificar se o firewall não modifica/corrompe os pacotes. Algumas diferenças esperadas são destacadas.



Pontos principais:

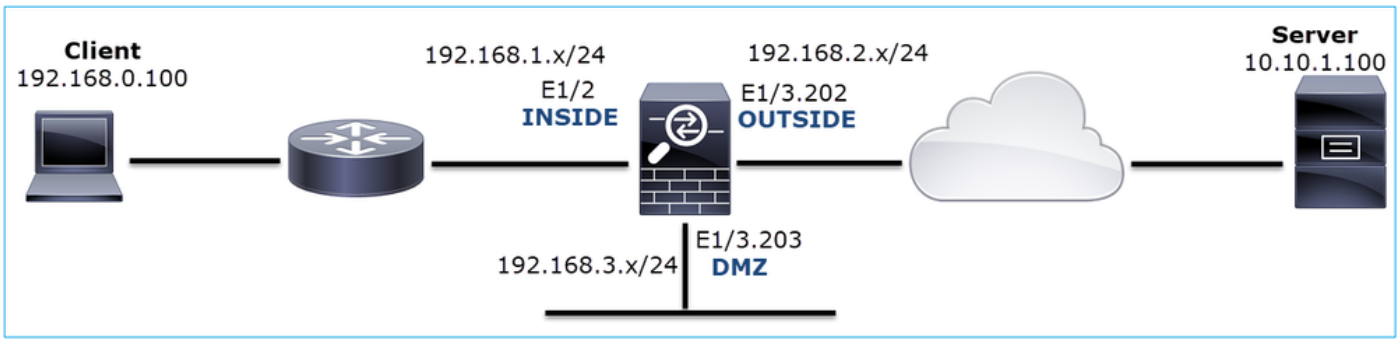
1. Os carimbos de data/hora são diferentes. Por outro lado, a diferença deve ser pequena e razoável. Isso depende dos recursos e das verificações de política aplicados ao pacote, bem como da carga no dispositivo.
2. O comprimento dos pacotes será diferente, especialmente se houver um cabeçalho dot1Q adicionado/removido pelo firewall apenas em um lado.
3. Os endereços MAC são diferentes.
4. Um cabeçalho dot1Q pode estar no lugar se a captura tiver sido feita em uma subinterface.
5. Os endereços IP são diferentes caso o NAT ou a conversão de endereço de porta (PAT) seja aplicada ao pacote.
6. As portas origem ou destino são diferentes caso o NAT ou o PAT sejam aplicados ao pacote.
7. Se você desabilitar a opção Relative Sequence Number do Wireshark, verá que os números de sequência/confirmação TCP são modificados pelo firewall devido à aleatoriedade do Initial Sequence Number (ISN).
8. Algumas opções TCP podem ser substituídas. Por exemplo, o firewall altera, por padrão, o tamanho máximo de segmento (MSS) do TCP para 1380, a fim de evitar a fragmentação de pacotes no caminho de trânsito.

Ação 3. Faça uma captura no destino.

Se possível, faça uma captura no próprio destino. Se isso não for possível, realize uma captura o mais perto possível do destino. O objetivo aqui é verificar quem envia o TCP RST (é o servidor de destino ou algum outro dispositivo no caminho?).

Caso 3. Handshake triplo do TCP + RST de um endpoint

Esta imagem mostra a topologia:



Descrição do problema: o HTTP não funciona

Fluxo afetado:

IP orig.: 192.168.0.100

IP do Horário de Verão: 10.10.1.100

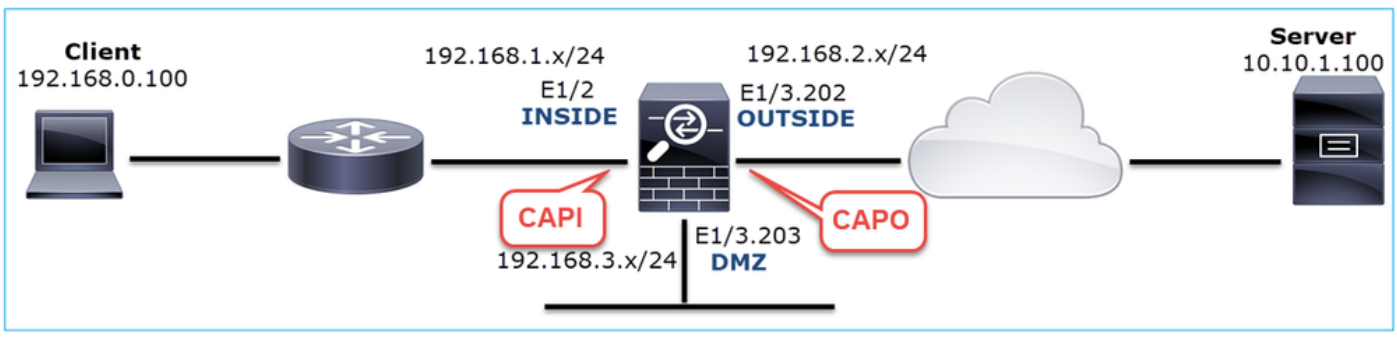
Protocolo: TCP 80

Capturar análise

Habilitar capturas no mecanismo LINA do FTD.

<#root>

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

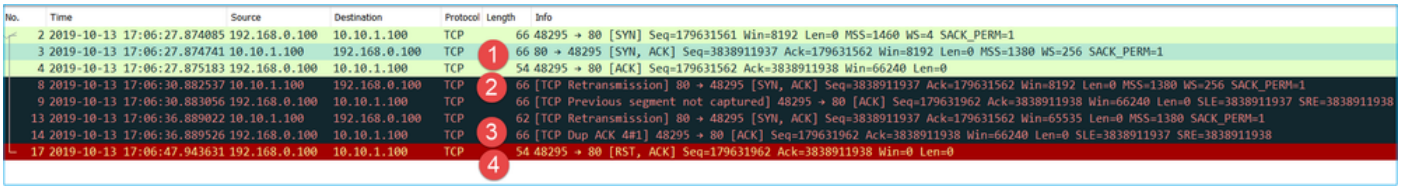


Capturas - cenário não funcional:

Há algumas maneiras diferentes pelas quais esse problema pode se manifestar em capturas.

3.1 - Handshake triplo do TCP + RST atrasado do cliente

O firewall captura CAPI e CAPO contêm os mesmos pacotes, como mostrado na imagem.



Pontos principais:

1. O handshake triplo do TCP passa pelo firewall.
2. O servidor retransmite o SYN/ACK.
3. O cliente retransmite o ACK.
4. Após ~20 seg, o cliente desiste e envia um TCP RST.

Ações recomendadas

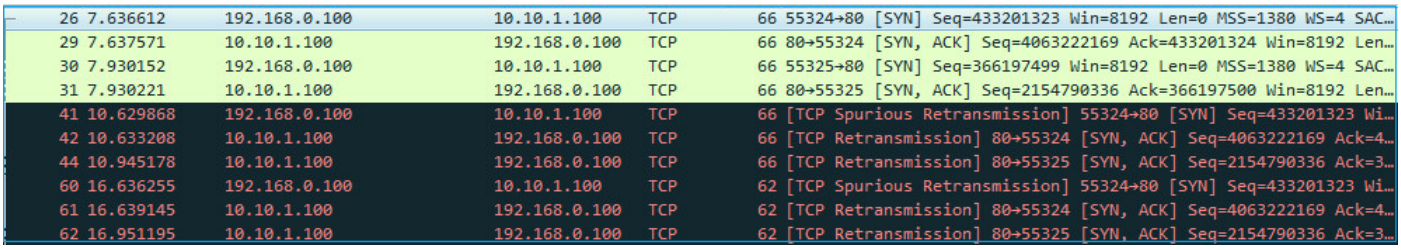
As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Faça capturas o mais perto possível dos dois endpoints.

As capturas de firewall indicam que o cliente ACK não foi processado pelo servidor. Isto baseia-se nos seguintes fatos:

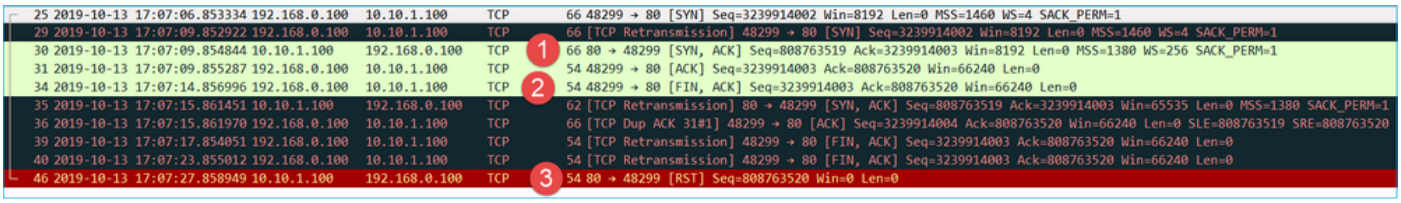
- O servidor retransmite o SYN/ACK.
- O cliente retransmite o ACK.
- O cliente envia um TCP RST ou FIN/ACK antes de qualquer dado.

A captura no servidor mostra o problema. O ACK do cliente do handshake triplo TCP nunca chegou:



3.2 - Handshake triplo do TCP + FIN/ACK atrasado do cliente + RST atrasado do servidor

O firewall captura CAPI e CAPO contêm os mesmos pacotes, como mostrado na imagem.



Pontos principais:

1. O handshake triplo do TCP passa pelo firewall.
2. Após ~5 s, o cliente envia um FIN/ACK.
3. Após ~20 seg, o servidor desiste e envia um TCP RST.

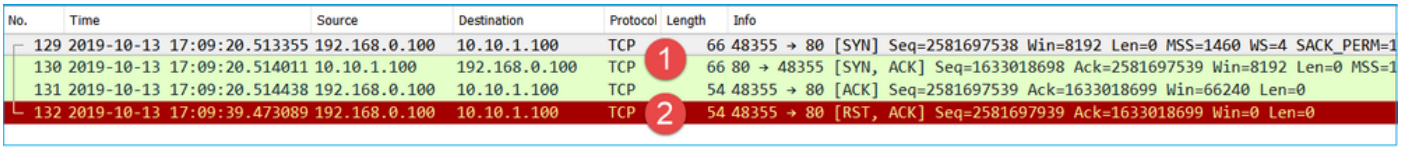
Com base nessa captura, pode-se concluir que, embora haja um handshake triplo do TCP através do firewall, parece que ele nunca é realmente concluído em um endpoint (as retransmissões indicam isso).

Ações recomendadas

Idêntico ao do caso 3.1

3.3 - Handshake triplo do TCP + RST atrasado do cliente

O firewall captura CAPI e CAPO contêm os mesmos pacotes, como mostrado na imagem.



Pontos principais:

1. O handshake triplo do TCP passa pelo firewall.
2. Após ~20 seg, o cliente desiste e envia um TCP RST.

Com base nestas capturas, pode concluir-se que:

- Após 5 a 20 segundos, um endpoint desiste e decide encerrar a conexão.

Ações recomendadas

Idêntico ao do caso 3.1

3.4 - Handshake triplo do TCP + RST imediato do servidor

O firewall captura o CAPI e o CAPO contêm esses pacotes, como mostrado na imagem.

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

Pontos principais:

1. O handshake triplo do TCP passa pelo firewall.
2. Há um TCP RST do servidor alguns milissegundos após o pacote ACK.

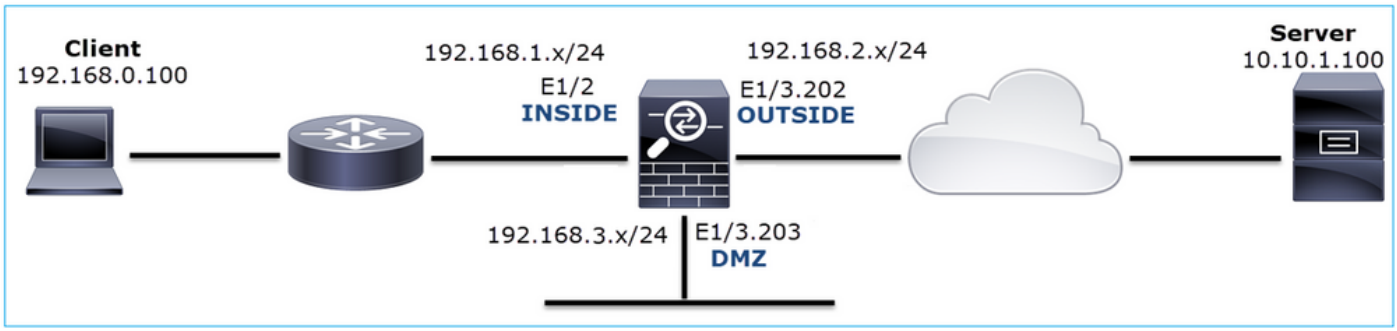
Ações recomendadas

Ação: Fazer capturas o mais próximo possível do servidor.

Um TCP RST imediato do servidor pode indicar um servidor com defeito ou um dispositivo no caminho que envia o TCP RST. Faça uma captura no próprio servidor e determine a origem do TCP RST.

Caso 4. TCP RST do cliente

Esta imagem mostra a topologia:



Descrição do problema: o HTTP não funciona.

Fluxo afetado:

IP orig.: 192.168.0.100

IP do Horário de Verão: 10.10.1.100

Protocolo: TCP 80

Capturar análise

Habilitar capturas no mecanismo LINA FTD.

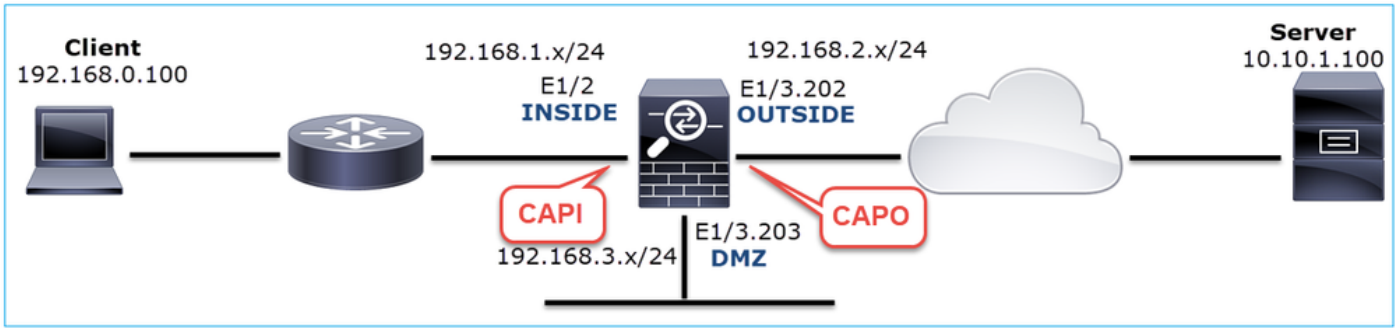
<#root>

```
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
```



Capturas - cenário não funcional:

Estes são os conteúdos 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
```

Estes são os conteúdos do CAPO:

<#root>

```
firepower#
```

```
show capture CAPO
```

```
11 packets captured
```

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

Os logs do firewall mostram:

<#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
```

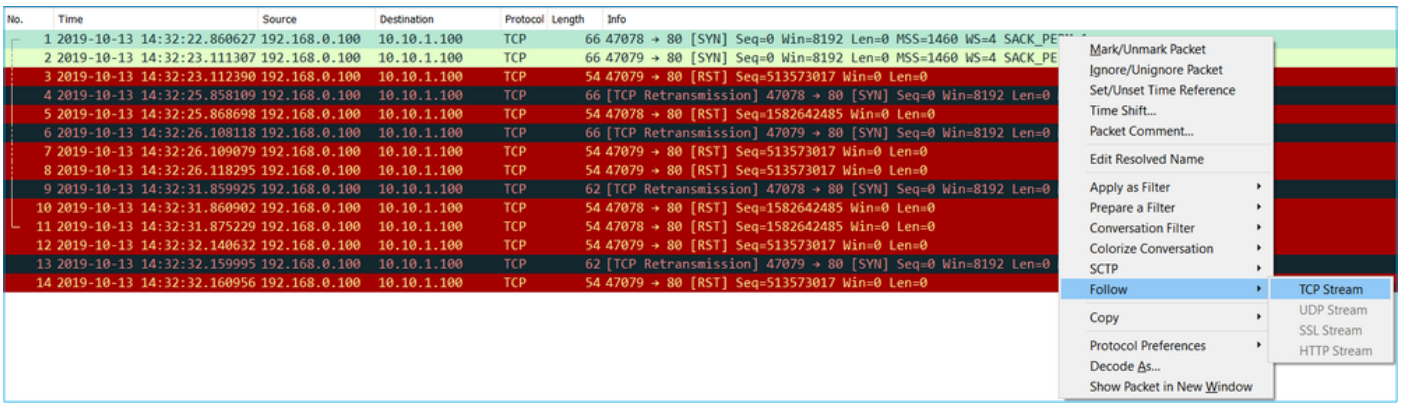
TCP Reset-O from INSIDE

```
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
```

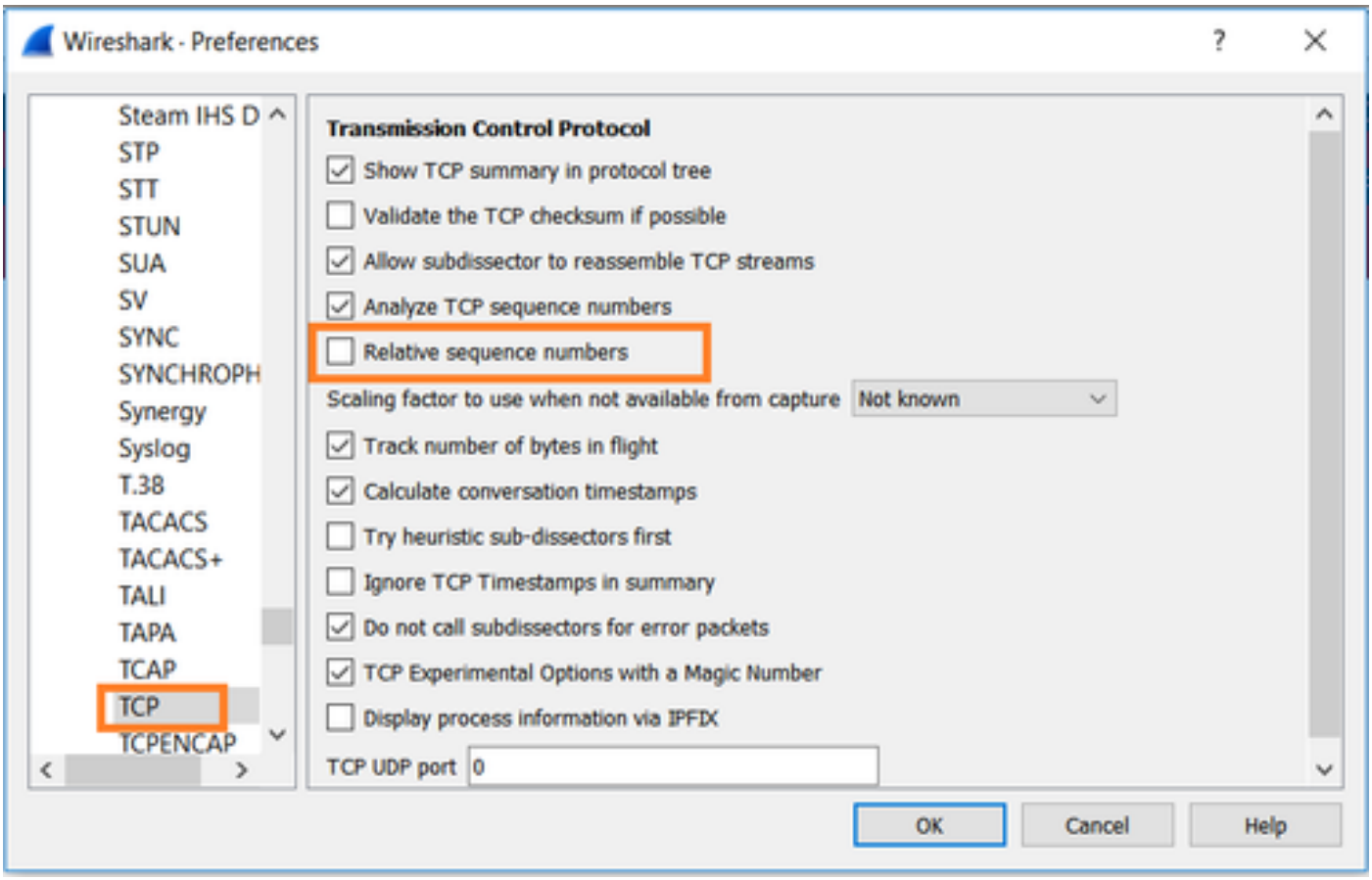
Esses logs indicam que há um TCP RST que chega à interface INSIDE do firewall

Captura CAPI no Wireshark:

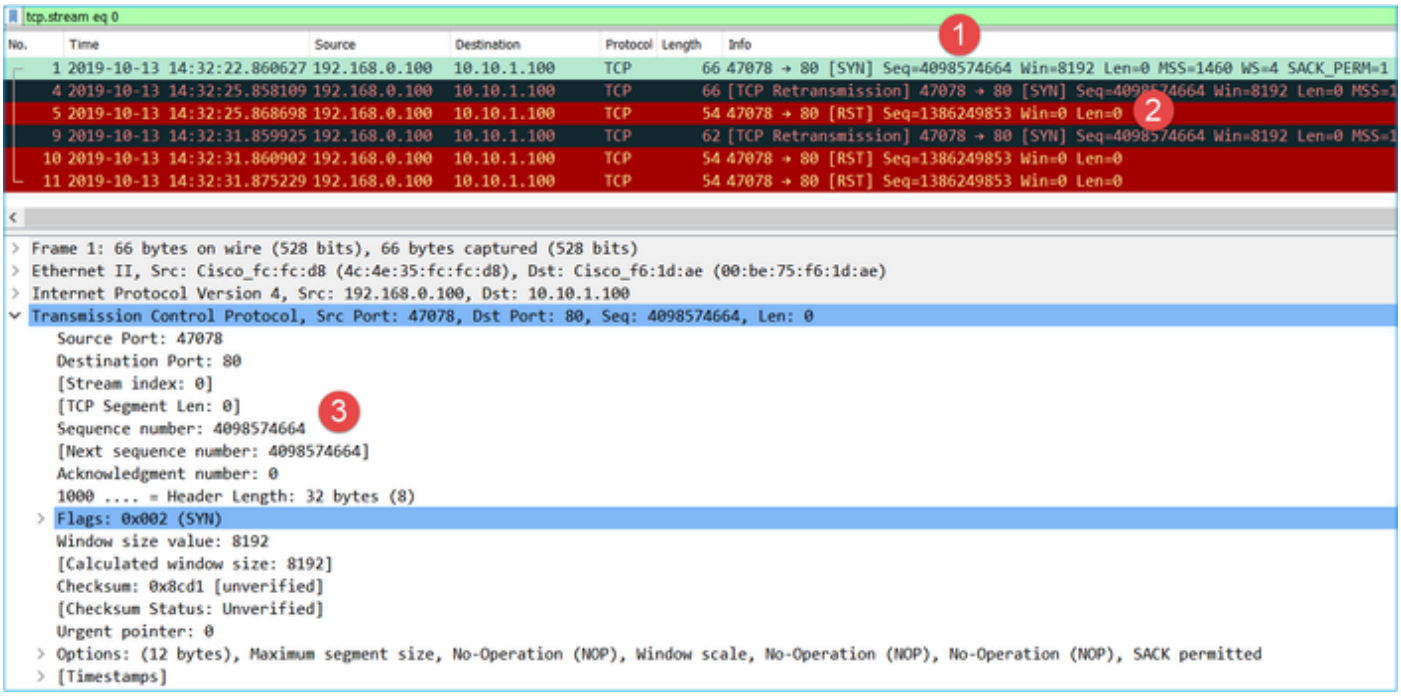
Siga o primeiro fluxo TCP, como mostrado na imagem.



Em Wireshark, navegue para Edit > Preferences > Protocols > TCP e desmarque a opção Relative sequence numbers como mostrado na imagem.



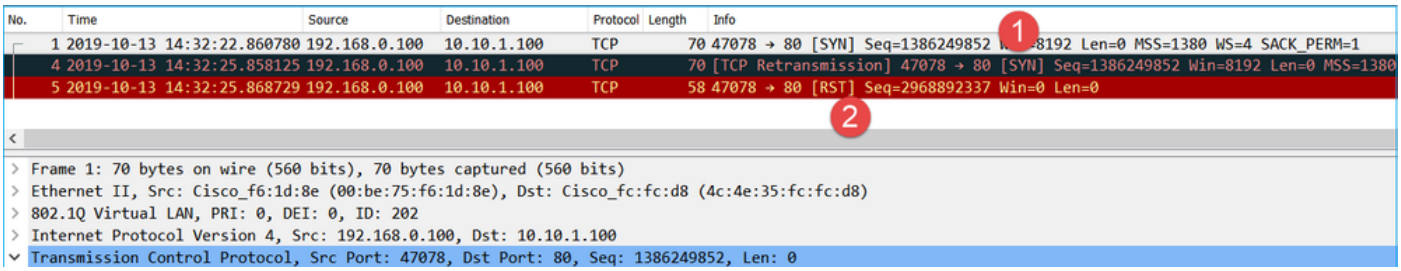
Esta imagem mostra o conteúdo do primeiro fluxo na captura CAPI:



Pontos principais:

1. O cliente envia um pacote TCP SYN.
2. O cliente envia um pacote TCP RST.
3. O pacote TCP SYN tem um valor de número de sequência igual a 4098574664.

O mesmo fluxo na captura CAPO contém:



Pontos principais:

1. O cliente envia um pacote TCP SYN. O firewall torna aleatório o ISN.
2. O cliente envia um pacote TCP RST.

Com base nas duas capturas, pode concluir-se que:

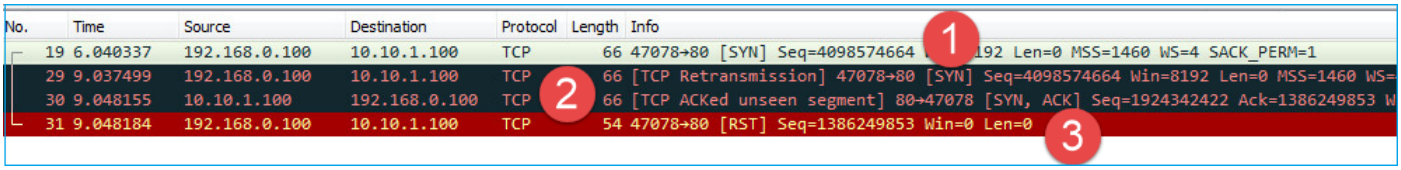
- Não há handshake triplo TCP entre o cliente e o servidor.
- Há um TCP RST que vem do cliente. O valor do número de sequência TCP RST na captura CAPI é 1386249853.

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Capture o cliente.

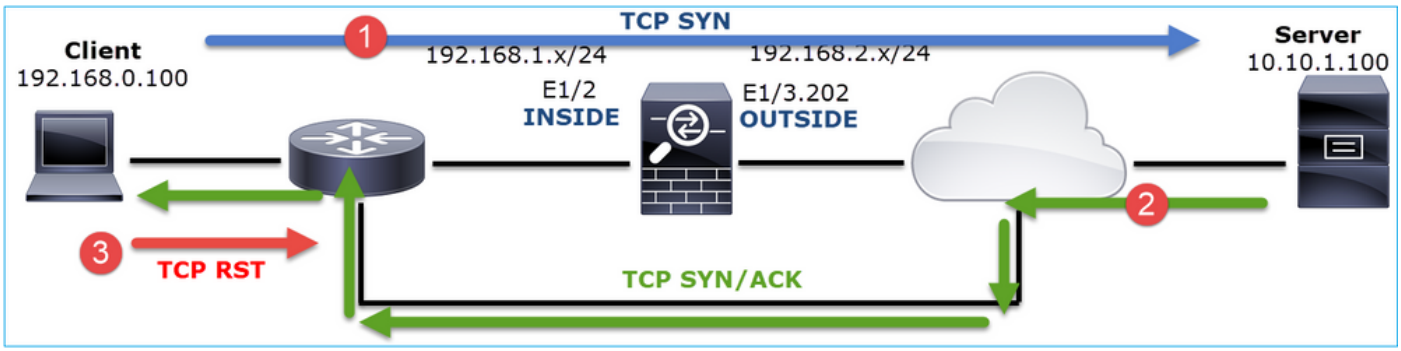
Com base nas capturas coletadas no firewall, há uma forte indicação de um fluxo assimétrico. Isso se baseia no fato de que o cliente envia um TCP RST com um valor de 1386249853 (o ISN aleatório):



Pontos principais:

1. O cliente envia um pacote TCP SYN. O número de sequência é 4098574664 e é o mesmo que o visto na interface INTERNA do firewall (CAPI)
2. Há um TCP SYN/ACK com número ACK 1386249853 (esperado devido à aleatorização ISN). Este pacote não foi visto nas capturas de firewall
3. O cliente envia um TCP RST, pois esperava um SYN/ACK com o valor de número ACK de 4098574665, mas recebeu o valor de 1386249853

Isso pode ser visualizado da seguinte maneira:

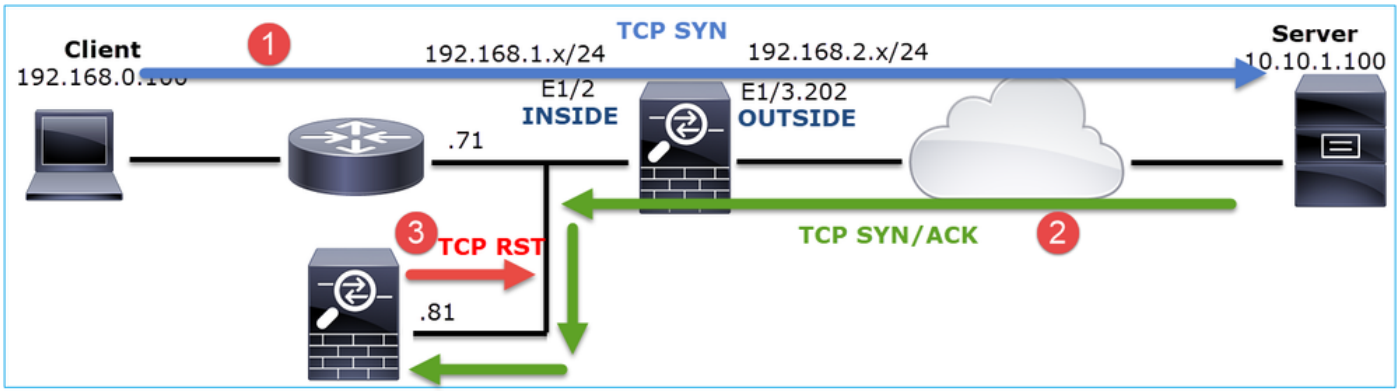


Ação 2. Verifique o roteamento entre o cliente e o firewall.

Confirme se:

- Os endereços MAC vistos nas capturas são os esperados.
- Certifique-se de que o roteamento entre o firewall e o cliente seja simétrico.

Há situações em que o RST vem de um dispositivo que fica entre o firewall e o cliente enquanto há um roteamento assimétrico na rede interna. Um caso típico é mostrado na imagem:



Nesse caso, a captura tem esse conteúdo. Observe a diferença entre o endereço MAC origem do pacote TCP SYN e o endereço MAC origem do TCP RST e o endereço MAC destino do pacote 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. Transferência TCP lenta (Cenário 1)

Descrição do problema:

A transferência SFTP entre os hosts 10.11.4.171 e 10.77.19.11 é lenta. Embora a largura de banda mínima (BW) entre os 2 hosts seja de 100 Mbps, a velocidade de transferência não vai além de 5 Mbps.

Ao mesmo tempo, a velocidade de transferência entre os hosts 10.11.2.124 e 172.25.18.134 é bem maior.

Material de Suporte:

A velocidade máxima de transferência para um único fluxo TCP é determinada pelo BDP (Bandwidth Delay Product). A fórmula usada é mostrada na imagem:

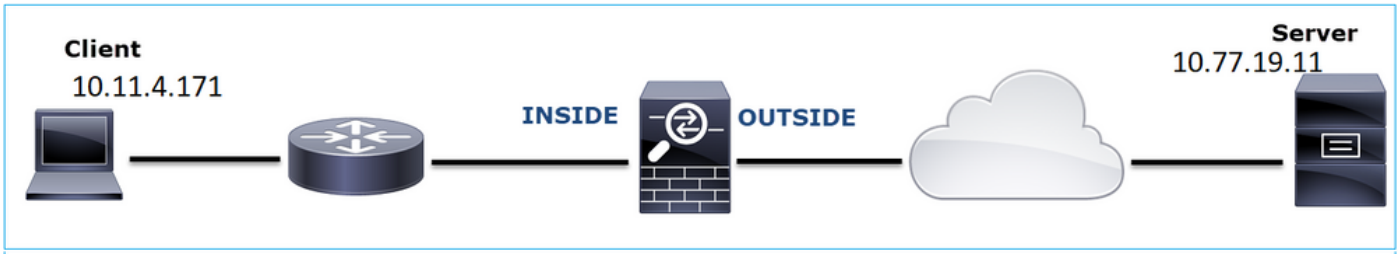
$$\text{Max Single TCP Flow Throughput [bps]} = \frac{\text{TCP Window (Bytes)}}{\text{RTT (Seconds)}} \times 8 \text{ [bits/Byte]}$$

Para obter mais detalhes sobre o BDP, verifique os recursos aqui:

- Por que seu aplicativo usa apenas 10 Mbps, mesmo que o link seja de 1 Gbps?
- BRKSEC-3021 - Avançado - Maximizando o desempenho do firewall

Cenário 1. Transferência lenta

Esta imagem mostra a topologia:



Fluxo afetado:

IP orig.: 10.11.4.171

IP do Horário de Verão: 10.77.19.11

Protocolo: SFTP (FTP sobre SSH)

Capturar análise

Habilitar capturas no mecanismo 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
```

Aviso: as capturas LINA em FP1xxx e FP21xx afetam a taxa de transferência de tráfego que passa pelo FTD. Não ative as capturas LINA nas plataformas FP1xxx e FP21xxx ao solucionar problemas de desempenho (transferência lenta através do FTD). Em vez disso, use o SPAN ou um dispositivo HW Tap além das capturas nos hosts origem e destino. O problema está documentado no bug da Cisco ID CSCvo30697.

<#root>

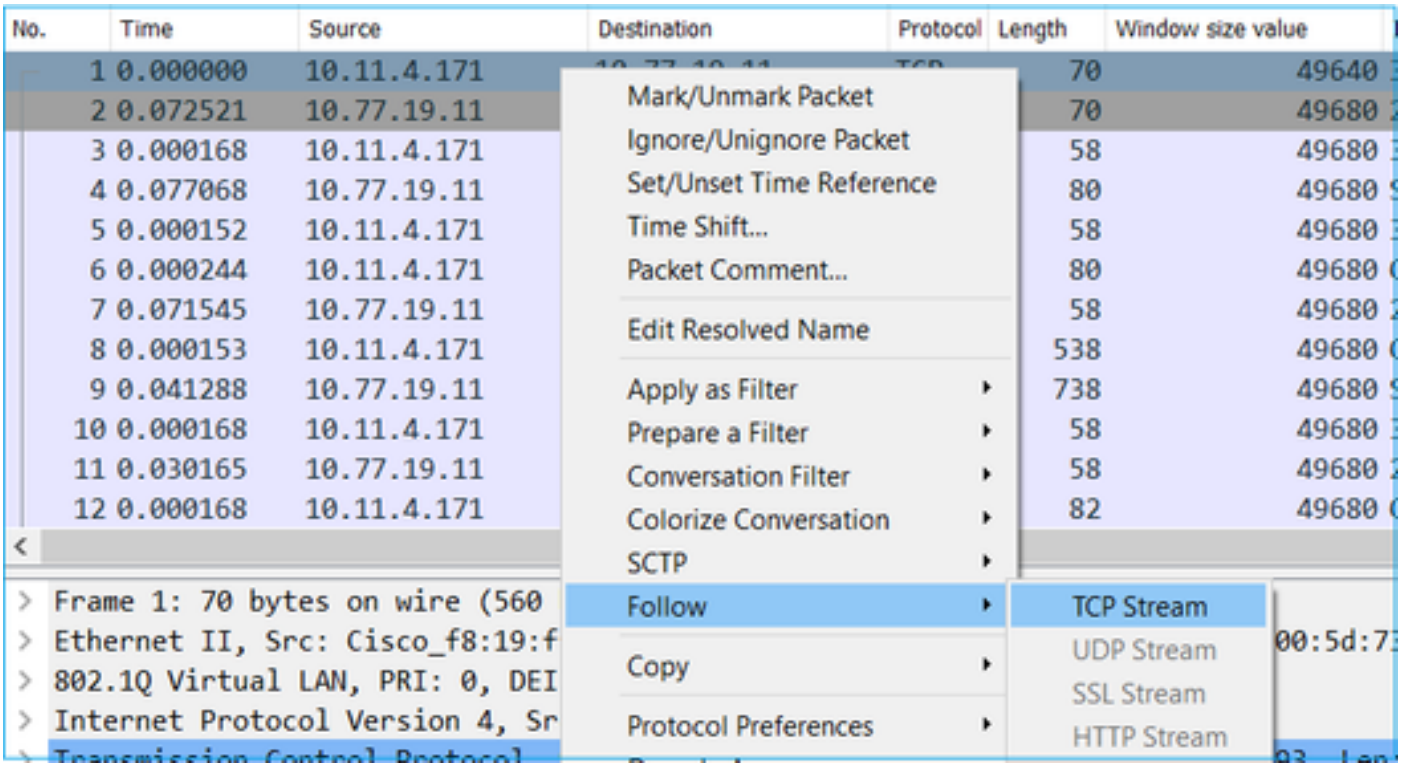
```
firepower#
capture CAPI type raw-data trace interface inside match icmp any any
WARNING: Running packet capture can have an adverse impact on performance.
```

Ações recomendadas

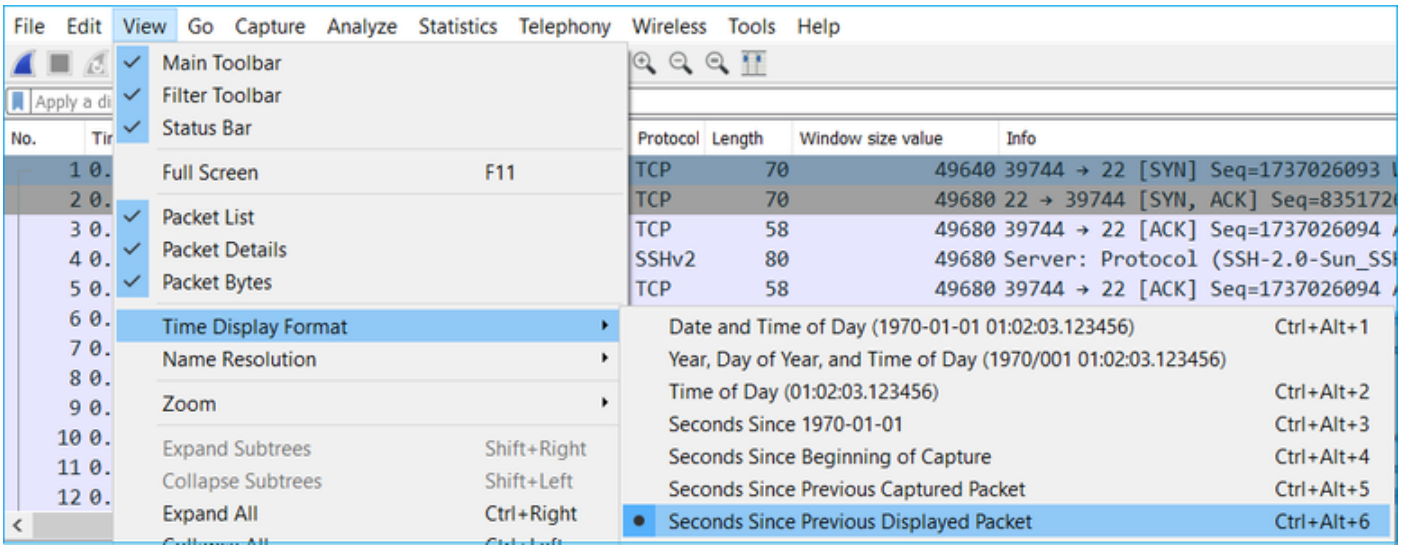
As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Cálculo do tempo de ida e volta (RTT)

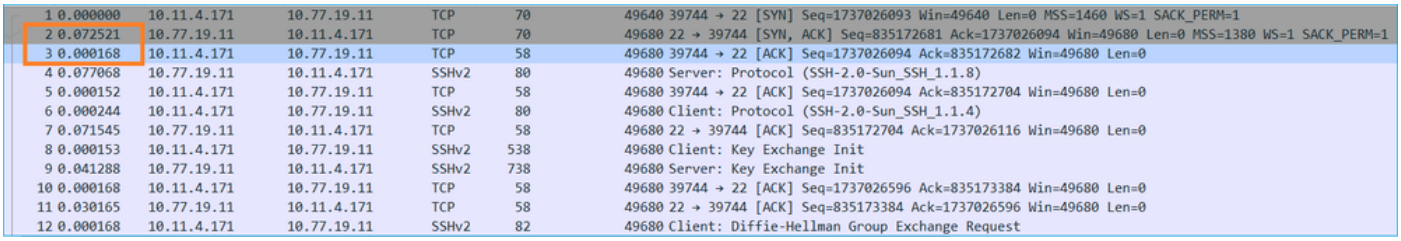
Primeiro, identifique o fluxo de transferência e siga-o:



Altere a visualização do Wireshark para mostrar Seconds Since the Previous Displayed Packet. Isso facilita o cálculo do RTT:



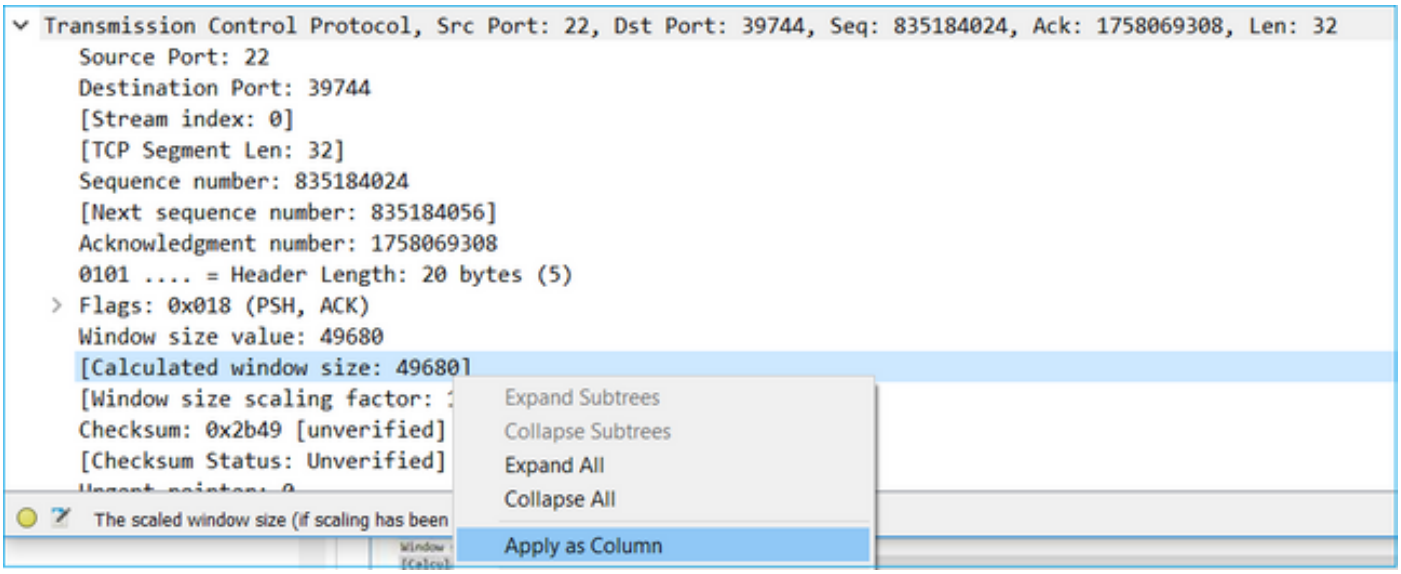
O RTT pode ser calculado pela adição dos valores de tempo entre 2 trocas de pacotes (um para a origem e um para o destino). Nesse caso, o pacote #2 mostra o RTT entre o firewall e o dispositivo que enviou o pacote SYN/ACK (servidor). O Packet #3 mostra o RTT entre o firewall e o dispositivo que enviou o pacote ACK (cliente). A adição dos 2 números fornece uma boa estimativa sobre o RTT fim-a-fim:



RTT ≈ 80 ms

Cálculo do Tamanho da Janela TCP

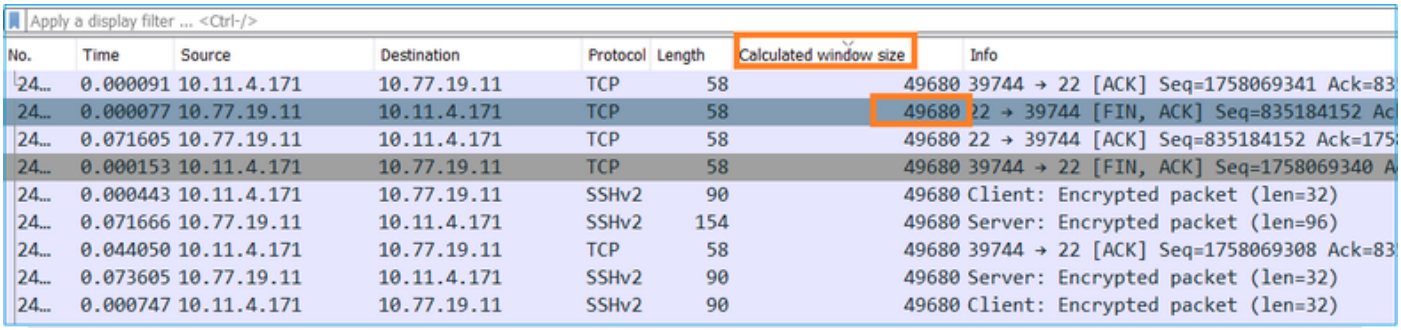
Expanda um pacote TCP, expanda o cabeçalho TCP, selecione Tamanho de janela calculado e selecione Aplicar como coluna:



Verifique a coluna Valor do tamanho da janela calculado para ver qual foi o valor do tamanho máximo da janela durante a sessão TCP. Você também pode selecionar o nome da coluna e classificar os valores.

Se você testar um download de arquivo (servidor > cliente), deverá verificar os valores anunciados pelo servidor. O valor do tamanho máximo da janela anunciado pelo servidor determina a velocidade máxima de transferência alcançada.

Nesse caso, o tamanho da janela TCP é ≈ 50000 Bytes

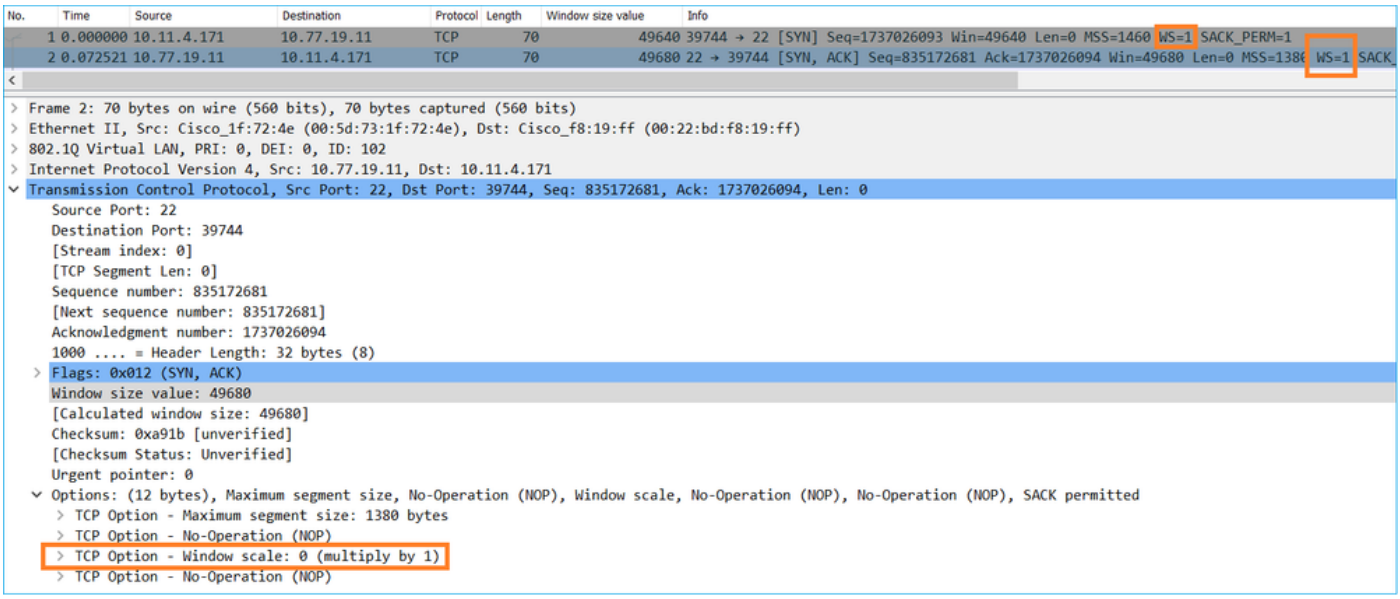


Com base nesses valores e com o uso da fórmula Bandwidth Delay Product, você obtém a largura de banda teórica máxima que pode ser obtida nessas condições: 50000*8/0.08 = largura de banda teórica máxima de 5 Mbps.

Isso corresponde às experiências do cliente neste caso.

Verifique cuidadosamente o handshake triplo do TCP. Ambos os lados, e mais importante o servidor, anunciam um valor de escala de janela de 0, o que significa 2^0 = 1 (sem escala de

janelas). Isso afeta negativamente a taxa de transferência:

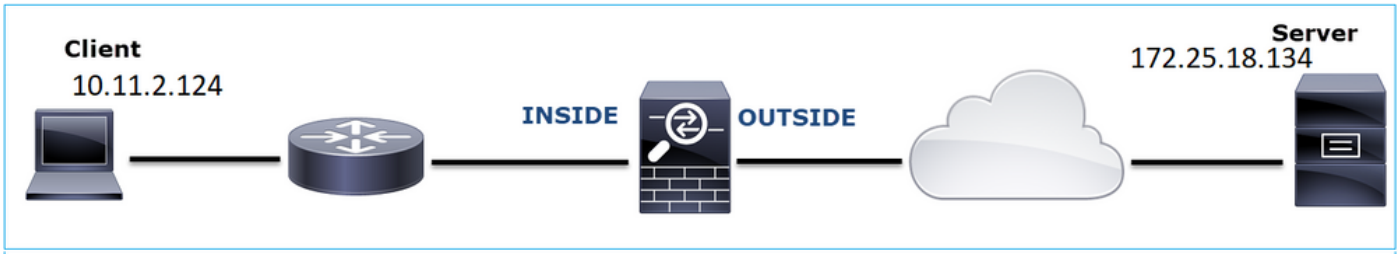


Neste ponto, é necessário fazer uma captura no servidor, confirmar que é ele que anuncia a escala de janela = 0 e reconfigurá-la (verifique a documentação do servidor para saber como fazer isso).

Cenário 2. Transferência rápida

Agora, vamos examinar o bom cenário (transferência rápida pela mesma rede):

Topologia:



O fluxo de interesse:

IP orig.: 10.11.2.124

IP do Horário de Verão: 172.25.18.134

Protocolo: SFTP (FTP sobre SSH)

Habilitar Capturas no mecanismo 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
```

Cálculo do Tempo de Ida e Volta (RTT): Neste caso, o RTT é ≈ 300 ms.

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

Cálculo do Tamanho da Janela TCP: O servidor anuncia um fator de escala de janela TCP de 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]
```

O tamanho da janela TCP do servidor é ≈ 1600000 Bytes:

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

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

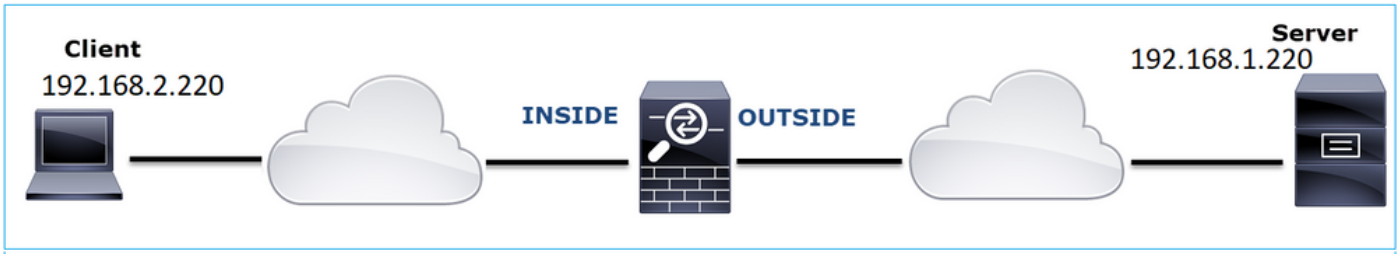
Com base nesses valores, a fórmula do Produto com Atraso de Largura de Banda fornece:

1600000*8/0.3 = velocidade de transferência teórica máxima de 43 Mbps

Caso 6. Transferência TCP lenta (Cenário 2)

Descrição do problema: a transferência de arquivos por FTP (download) pelo firewall está lenta.

Esta imagem mostra a topologia:



Fluxo afetado:

IP orig.: 192.168.2.220

IP do Horário de Verão: 192.168.1.220

Protocolo: FTP

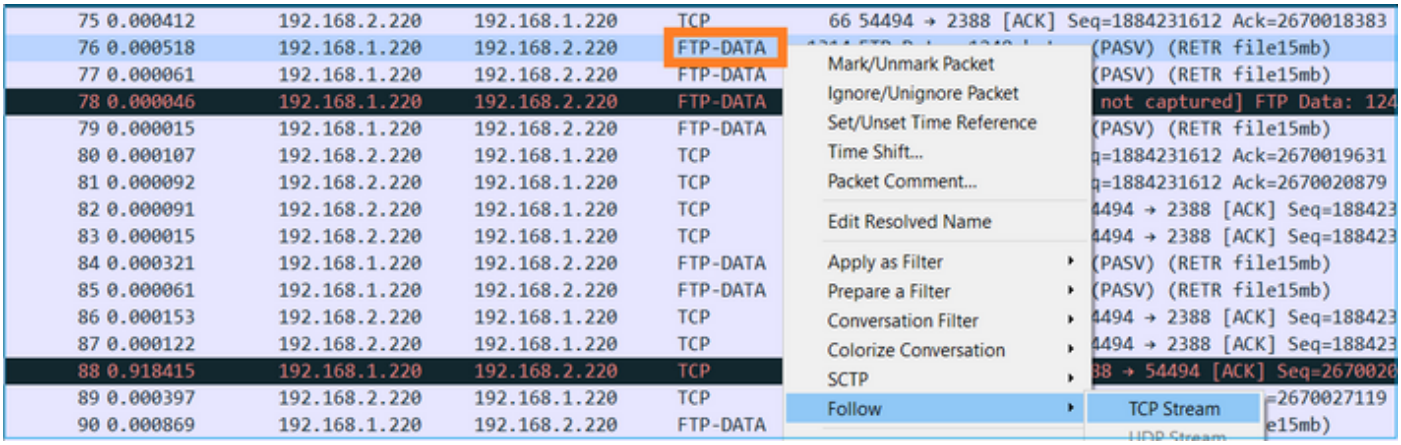
Capturar análise

Habilitar capturas no mecanismo LINA do FTD.

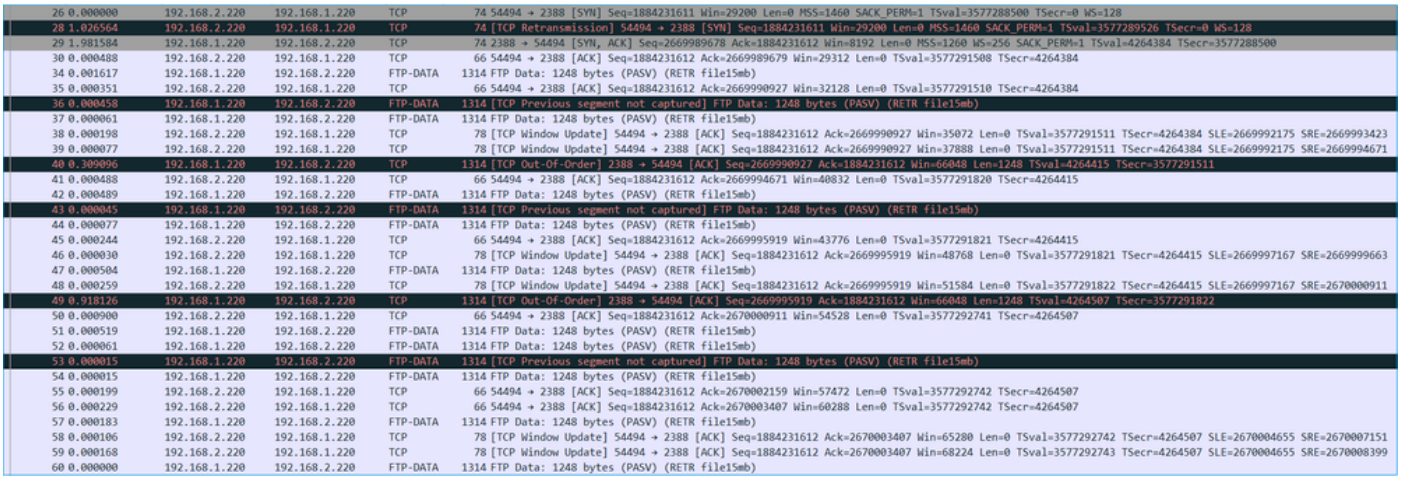
<#root>

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

Selecione um pacote FTP-DATA e siga o FTP Data Channel na captura FTD INSIDE (CAPI):

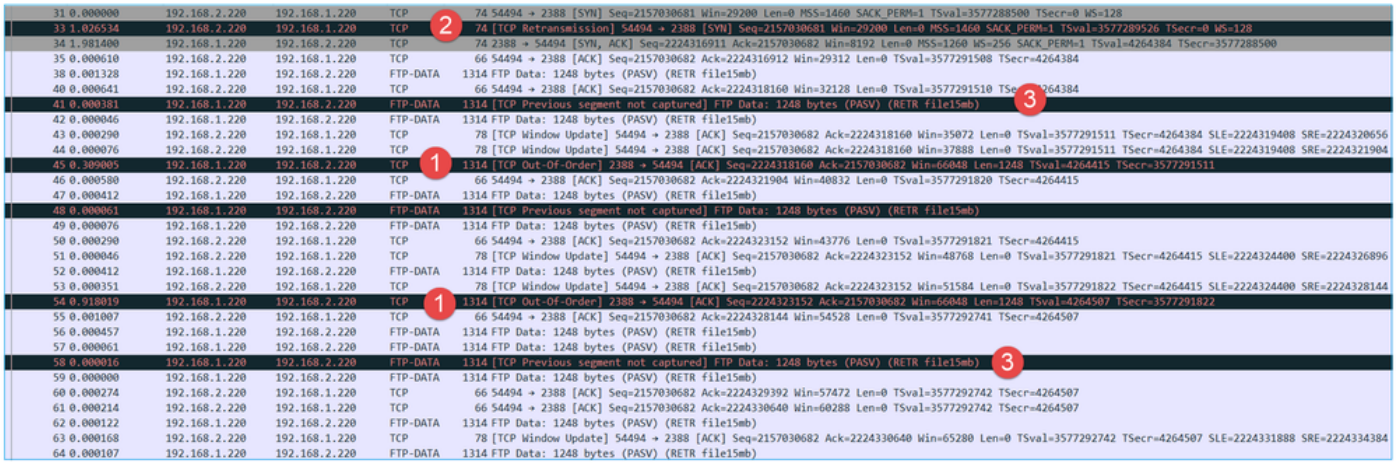


O conteúdo do fluxo FTP-DATA:



26	0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
28	1.026564	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
29	1.981584	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2669989678 Ack=1884231612 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
30	0.000488	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669989679 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
34	0.001617	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
35	0.000351	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
36	0.000458	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
37	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
38	0.000198	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=2669992175 SRE=2669993423
39	0.000077	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=2669992175 SRE=2669994671
40	0.309096	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669990927 Ack=1884231612 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
41	0.000488	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669994671 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
42	0.000489	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
43	0.000045	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
44	0.000077	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
45	0.000244	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
46	0.000030	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=2669997167 SRE=2669999663
47	0.000504	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
48	0.000259	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2669997167 SRE=2670000911
49	0.918126	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669995919 Ack=1884231612 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
50	0.000900	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670000911 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
51	0.000519	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
52	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
54	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
55	0.000199	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670002159 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
56	0.000229	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
57	0.000183	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58	0.000106	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=65280 Len=0 TSval=3577292742 TSecr=4264507 SLE=2670004655 SRE=2670007151
59	0.000168	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=68224 Len=0 TSval=3577292743 TSecr=4264507 SLE=2670004655 SRE=2670008399
60	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

O conteúdo de captura CAPO:

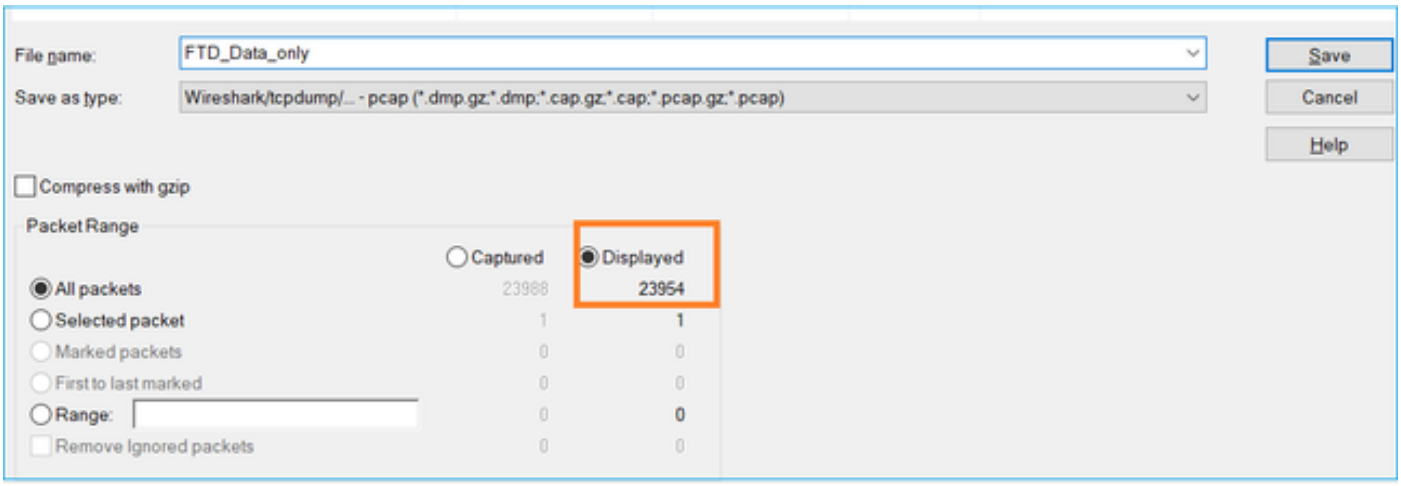


31	0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
33	1.026534	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 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=2224319408 SRE=2224321904
45	0.309005	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224318160 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
46	0.000580	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
47	0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
48	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
49	0.000076	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
50	0.000290	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
51	0.000046	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=2224324400 SRE=2224326896
52	0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53	0.000351	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2224324400 SRE=2224328144
54	0.918019	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224323152 Ack=2157030682 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
55	0.001007	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224328144 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
56	0.000457	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
57	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58	0.000016	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
59	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
60	0.000274	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224329392 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
61	0.000214	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224330640 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
62	0.000122	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
63	0.000168	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224330640 Win=65280 Len=0 TSval=3577292742 TSecr=4264507 SLE=2224331888 SRE=2224334384
64	0.000107	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

Pontos principais:

1. Há pacotes TCP fora de ordem (OOO).
2. Há uma retransmissão de TCP.
3. Há uma indicação de perda de pacotes (pacotes descartados).

Dica: salve as capturas enquanto navega para Arquivo > Exportar pacotes especificados. Em seguida, salve apenas o intervalo de pacotes Exibido.



Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Identifique o local de perda de pacotes.

Em casos como esse, você deve fazer capturas simultâneas e usar a metodologia divide and conquer para identificar os segmentos de rede que causam a perda de pacotes. Do ponto de vista do firewall, há três cenários principais:

1. A perda de pacotes é causada pelo próprio firewall.
2. A perda de pacotes é causada por downstream para o dispositivo de firewall (direção do servidor para o cliente).
3. A perda de pacotes é causada no upstream para o dispositivo de firewall (direção do cliente para o servidor).

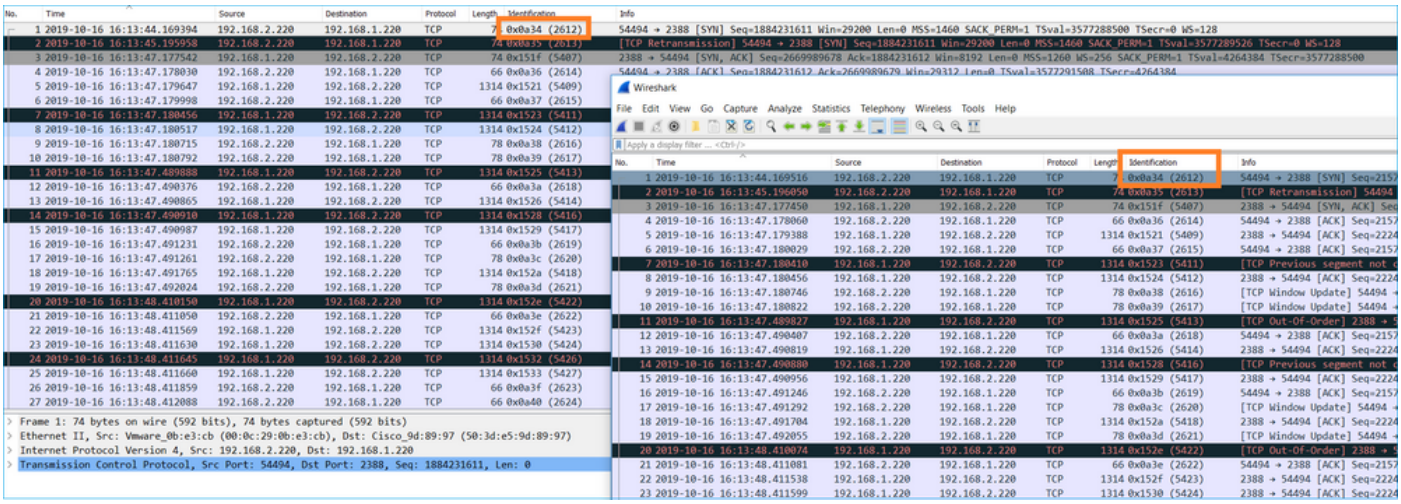
Perda de pacotes causada pelo firewall: para identificar se a perda de pacotes é causada pelo firewall, é necessário comparar a captura de entrada com a captura de saída. Há muitas maneiras de comparar 2 capturas diferentes. Esta seção demonstra uma maneira de fazer essa tarefa.

Procedimento para Comparar 2 Capturas para Identificar a Perda de Pacotes

Etapa 1. Certifique-se de que as 2 capturas contenham pacotes da mesma janela de tempo. Isso significa que não deve haver pacotes em uma captura que foram capturados antes ou depois da outra captura. Há algumas maneiras de fazer isso:

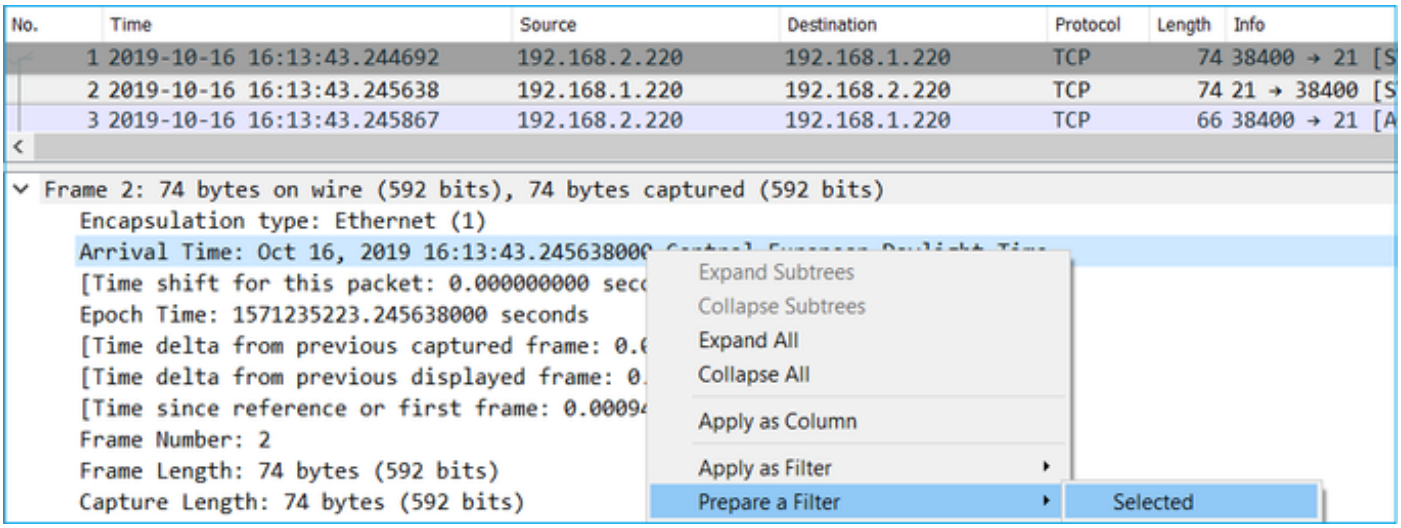
- Verifique o primeiro e o último valores de identificação (ID) IP do pacote.
- Verifique o primeiro e o último valores de timestamp do pacote.

Neste exemplo, você pode ver que os primeiros pacotes de cada captura têm os mesmos valores de ID IP:



Caso não sejam os mesmos:

1. Compare os Timestamps do primeiro pacote de cada captura.
2. Na captura com o Timestamp mais recente, obtenha um filtro dele. Altere o filtro Timestamp de == para >= (o primeiro pacote) e <= (o último pacote), por exemplo:



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

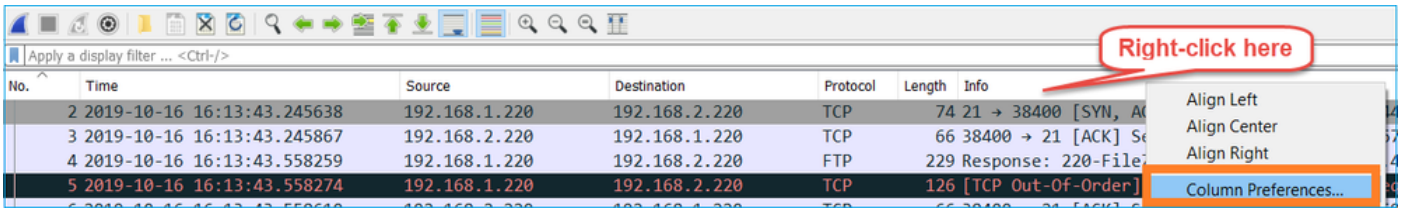
3. Exporte os pacotes especificados para uma nova captura, selecione Arquivo > Exportar Pacotes Especificados e salve os pacotes Exibidos. Nesse ponto, as duas capturas devem conter pacotes que cubram a mesma janela de tempo. Agora você pode iniciar a comparação das 2 capturas.

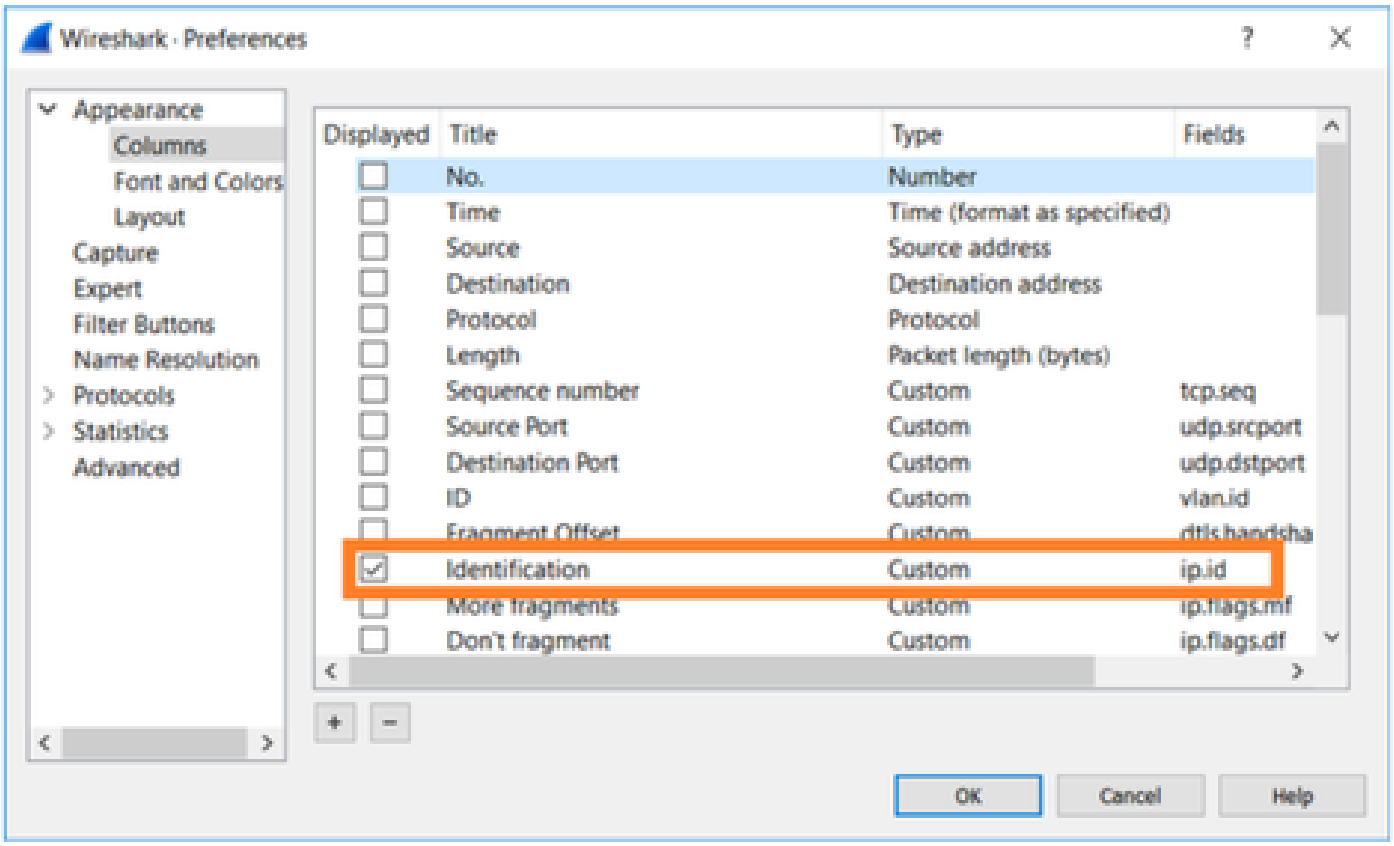
Etapa 2. Especifique qual campo de pacote é usado para a comparação entre as 2 capturas. Exemplo de campos que podem ser usados:

- Identificação IP
- Número de Sequência RTP
- Número de sequência ICMP

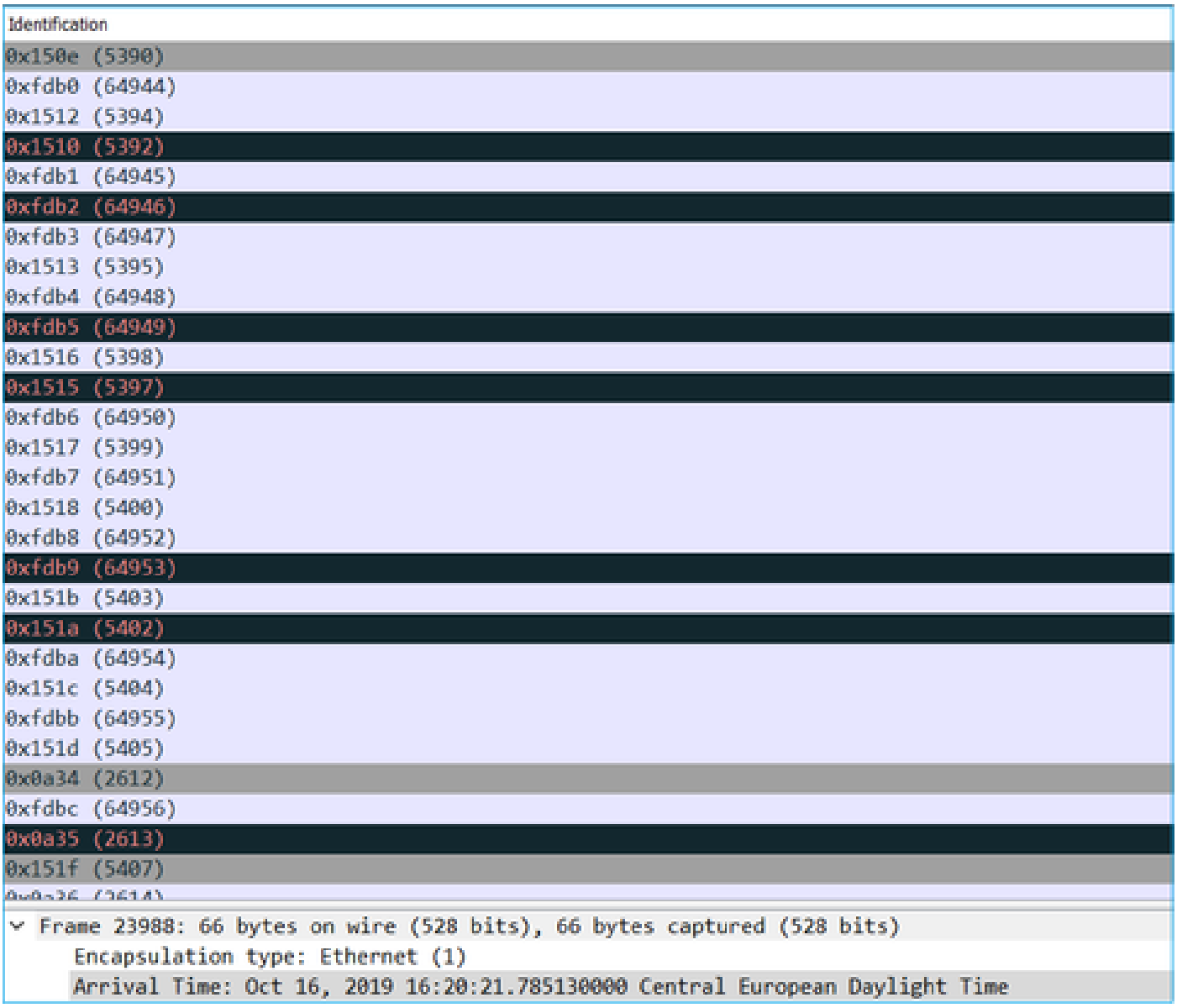
Crie uma versão de texto de cada captura que contenha o campo para cada pacote especificado

na etapa 1. Para fazer isso, deixe apenas a coluna de interesse, por exemplo, se quiser comparar pacotes com base na identificação de IP, modifique a captura conforme mostrado na imagem.

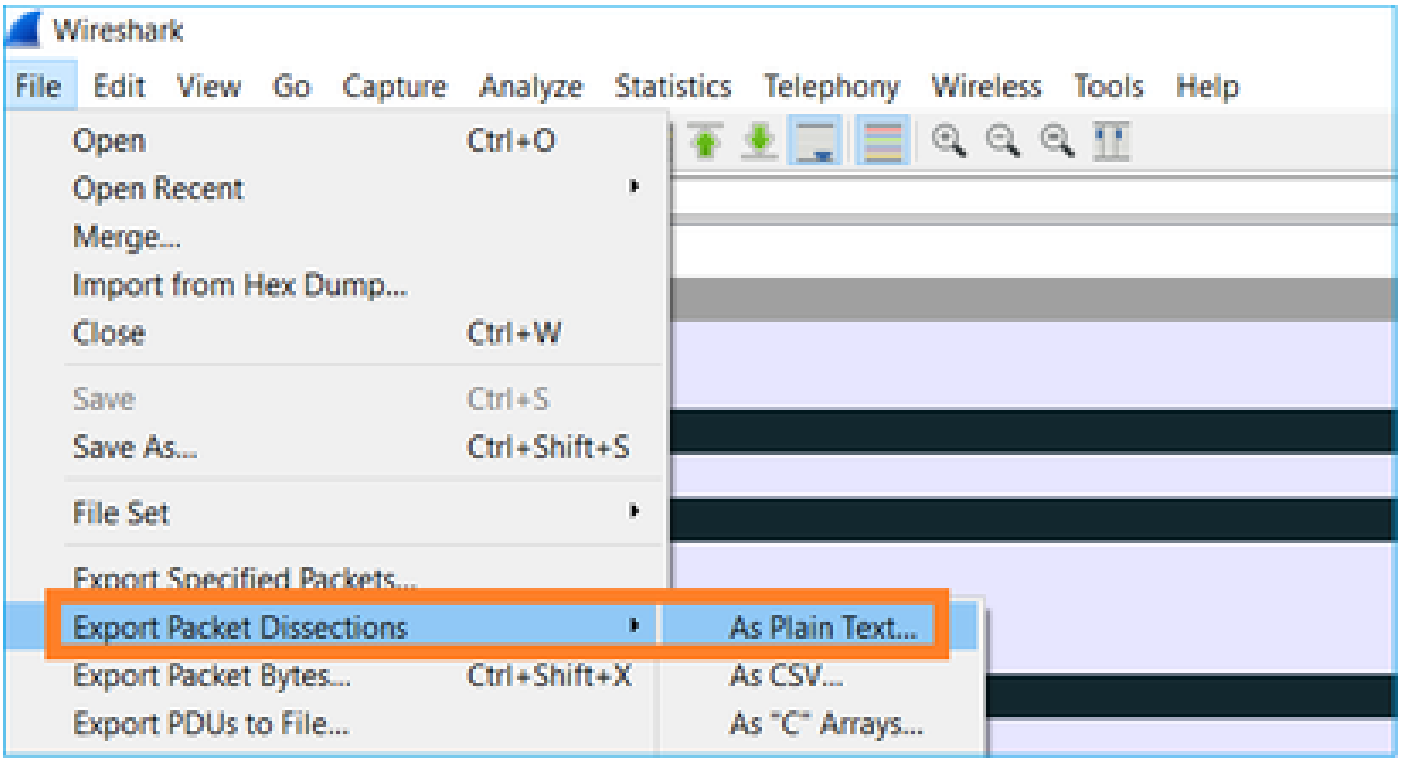




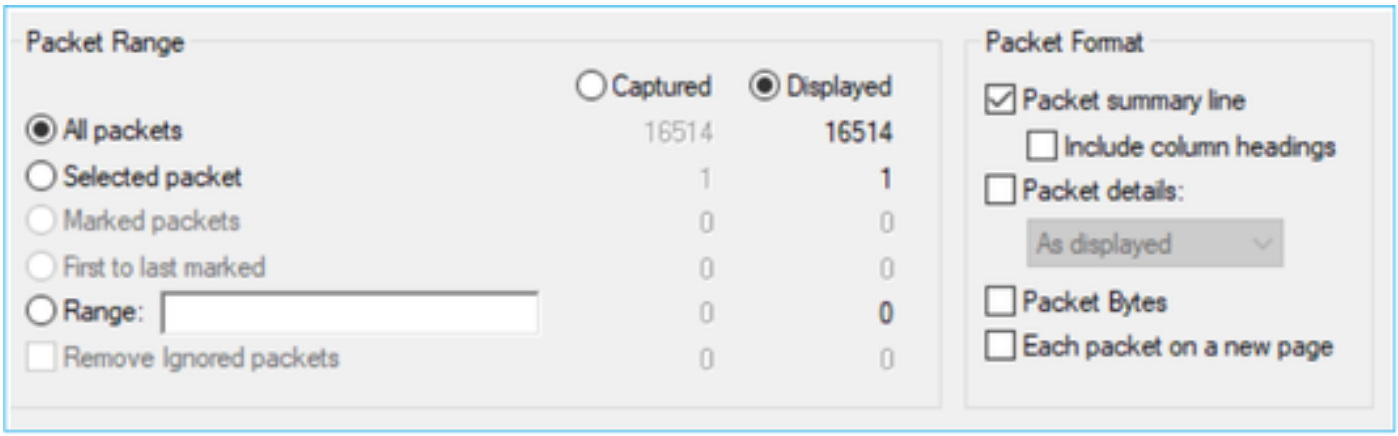
O resultado:



Etapa 3. Crie uma versão de texto da captura (Arquivo > Exportar Disseções de Pacote > Como Texto sem Formatação...), conforme mostrado na imagem:



Desmarque as opções Incluir títulos de coluna e Detalhes do pacote para exportar apenas os valores do campo exibido, como mostrado na imagem:



Etapa 4. Classifique os pacotes nos arquivos. Você pode usar o comando Linux sort para fazer isso:

<#root>

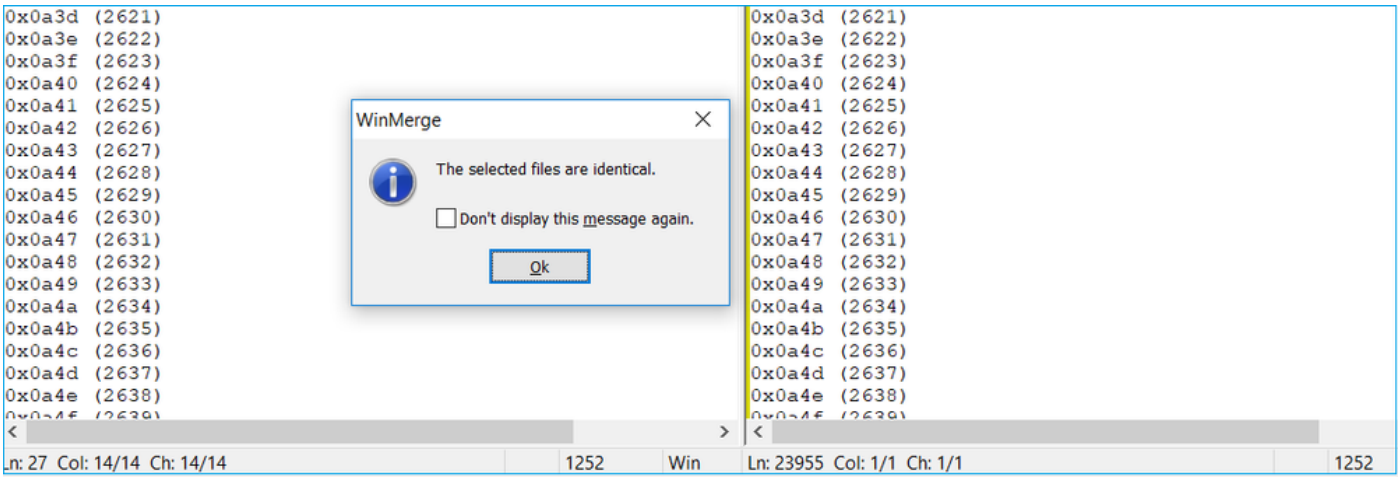
#

sort CAPI_IDs > file1.sorted

#

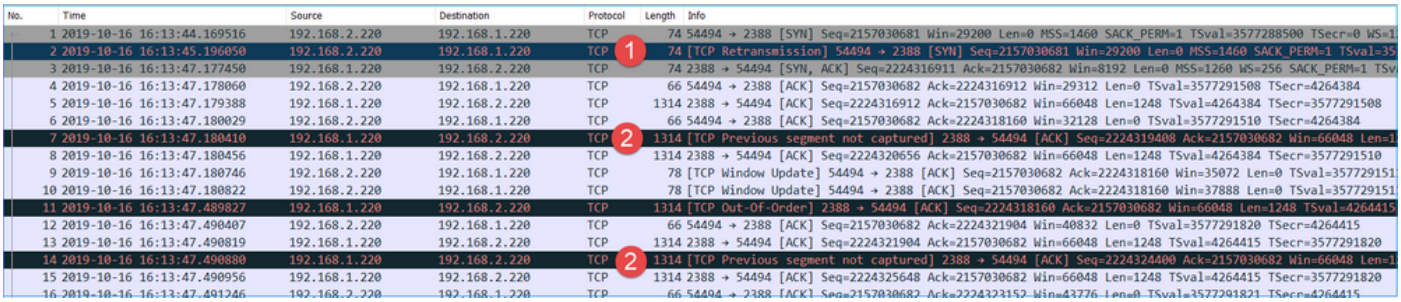
sort CAPO_IDs > file2.sorted

Etapa 5. Use uma ferramenta de comparação de texto (por exemplo, WinMerge) ou o comando Linux diff para encontrar as diferenças entre as 2 capturas.



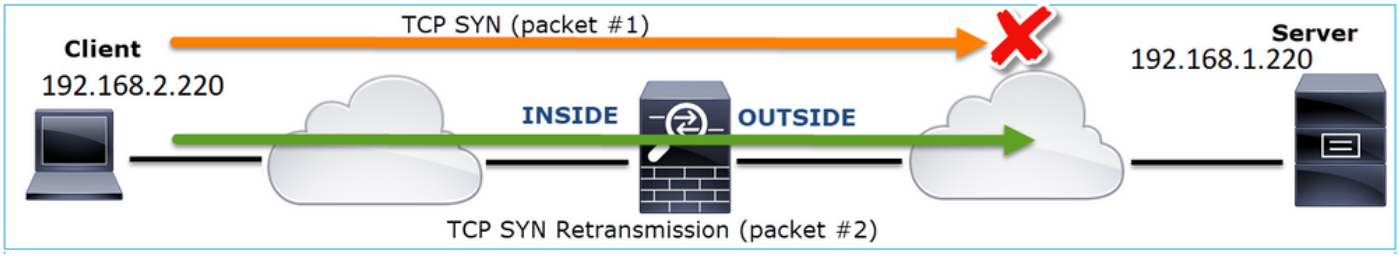
Nesse caso, as capturas CAPI e CAPO para o tráfego de dados FTP são idênticas. Isso prova que a perda de pacotes não foi causada pelo firewall.

Identificar perda de pacotes upstream/downstream.

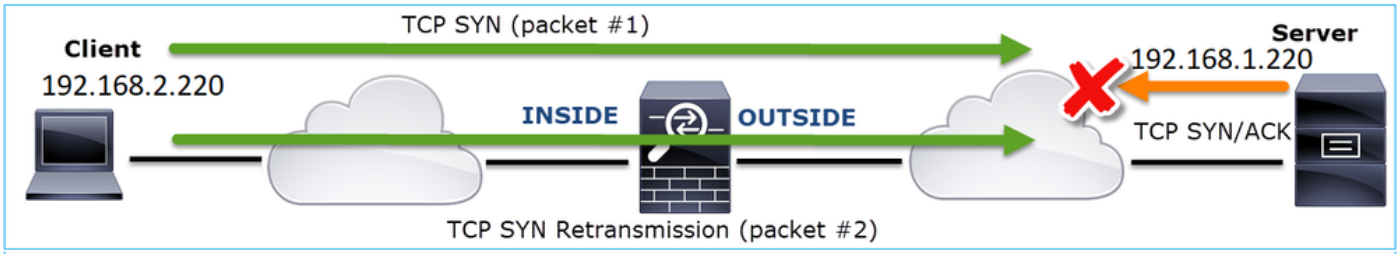


Pontos principais:

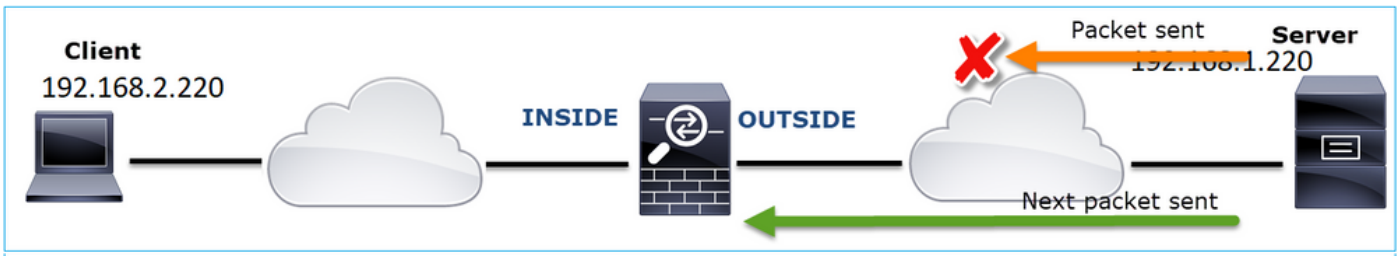
1. Este pacote é uma Retransmissão TCP. Especificamente, é um pacote TCP SYN enviado do cliente para o servidor para Dados FTP em Modo Passivo. Como o cliente reenvia o pacote e você pode ver o SYN inicial (#1 do pacote), o pacote foi perdido upstream para o firewall.



Nesse caso, existe a possibilidade de que o pacote SYN tenha chegado ao servidor, mas o pacote SYN/ACK tenha sido perdido no caminho de volta:



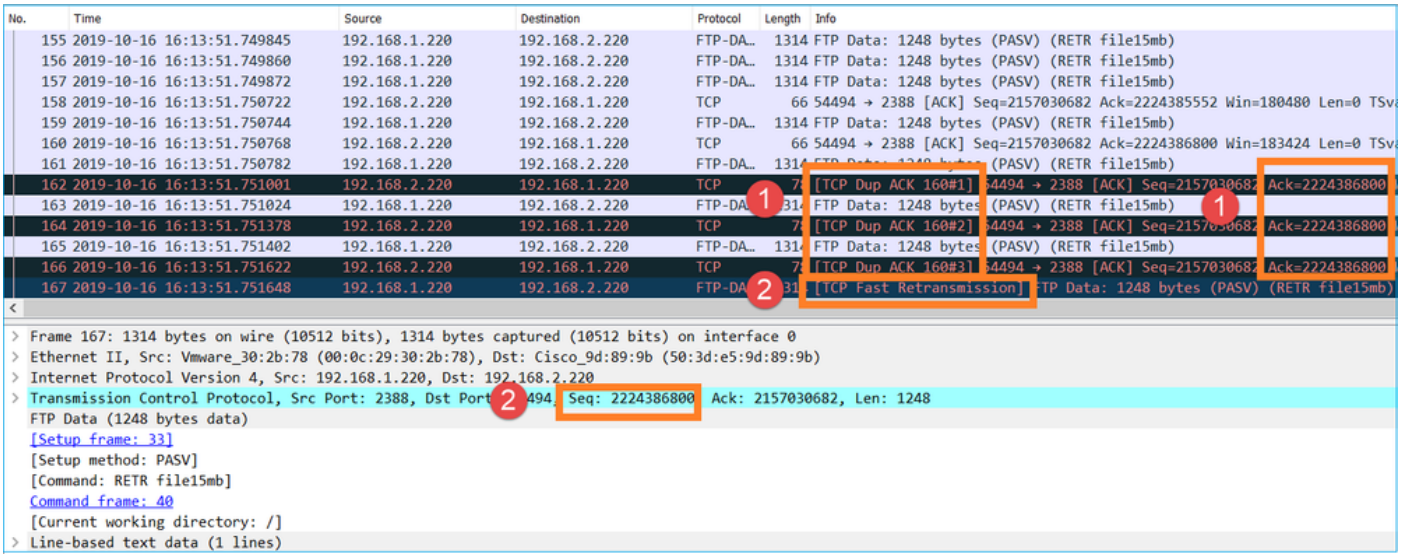
2. Há um pacote do servidor e o Wireshark identificou que o segmento anterior não foi visto/capturado. Como o pacote não capturado foi enviado do servidor para o cliente e não foi visto na captura do firewall, isso significa que o pacote foi perdido entre o servidor e o firewall.



Isso indica que há perda de pacotes entre o servidor FTP e o firewall.

Ação 2. Faça capturas adicionais.

Faça capturas adicionais junto com as capturas nos endpoints. Tente aplicar o método divide and conquer para isolar ainda mais o segmento problemático que causa a perda de pacotes.



Pontos principais:

1. O receptor (o cliente FTP, nesse caso) rastreia os números de sequência TCP recebidos. Se detectar que um pacote foi perdido (um número de sequência esperado foi ignorado), ele gerará um pacote ACK com o ACK='expected sequence number that was skipped'. Neste exemplo, Ack=2224386800.

2. O ACK Dup dispara uma retransmissão rápida de TCP (retransmissão dentro de 20 ms depois que um ACK Duplicado é recebido).

O que significam ACKs duplicados?

- Alguns ACKs duplicados, mas nenhuma retransmissão real, indicam que é mais provável que existam pacotes que cheguem fora de ordem.
- ACKs duplicados seguidos de retransmissões reais indicam que há alguma quantidade de perda de pacotes.

Ação 3. Calcule o tempo de processamento do firewall para pacotes de trânsito.

Aplique a mesma captura em 2 interfaces diferentes:

<#root>

```
firepower#
capture CAPI buffer 33554432 interface INSIDE match tcp host 192.168.2.220 host 192.168.1.220

firepower#
capture CAPI interface OUTSIDE
```

Exportar a captura verifica a diferença de tempo entre os pacotes de entrada vs de saída

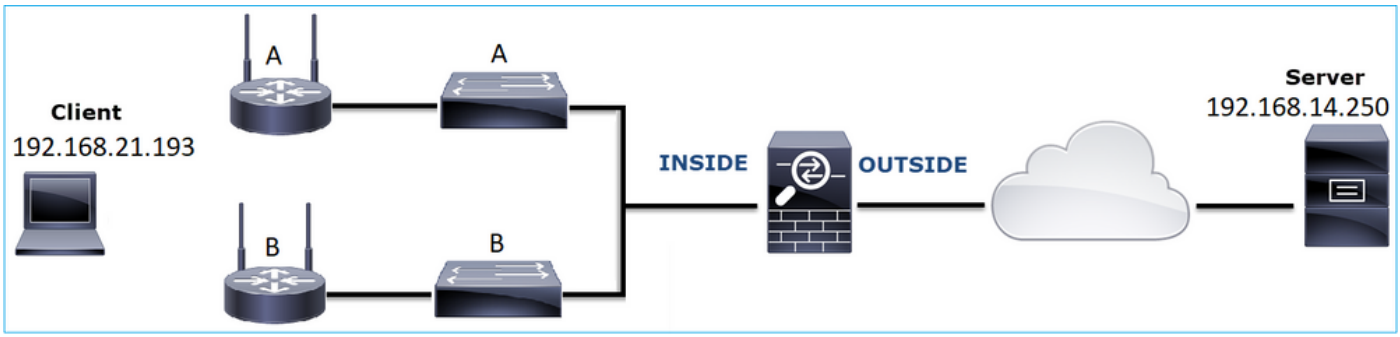
Caso 7. Problema de conectividade de TCP (Corrupção de pacote)

Descrição do problema:

O cliente sem fio (192.168.21.193) tenta se conectar a um servidor de destino (192.168.14.250 - HTTP) e há dois cenários diferentes:

- Quando o cliente se conecta ao Ponto de Acesso (AP) 'A', a conexão HTTP não funciona.
- Quando o cliente se conecta ao Ponto de Acesso (AP) 'B', a conexão HTTP funciona.

Esta imagem mostra a topologia:



Fluxo afetado:

IP orig.: 192.168.21.193

IP do Horário de Verão: 192.168.14.250

Protocolo: TCP 80

Capturar análise

Habilitar capturas no mecanismo 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
```

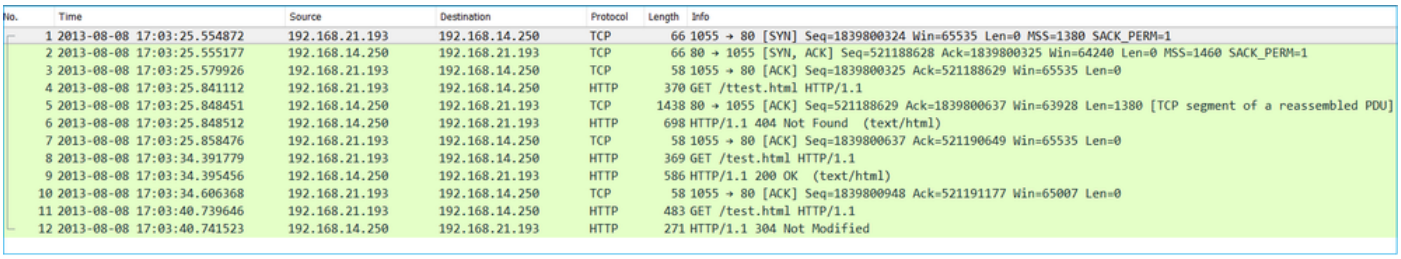
Capturas - Cenário Funcional:

Como parâmetro, é sempre muito útil ter capturas de um cenário em boas condições.

Esta imagem mostra a captura realizada na interface INSIDE do NGFW

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 17:03:25.554582	192.168.21.193	192.168.14.250	TCP	66	1055 → 80 [SYN] Seq=1341231 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 17:03:25.555238	192.168.14.250	192.168.21.193	TCP	66	80 → 1055 [SYN, ACK] Seq=1015787006 Ack=1341232 Win=64240 Len=0 MSS=1380 SACK_PERM=1
3	2013-08-08 17:03:25.579910	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1341232 Ack=1015787007 Win=65535 Len=0
4	2013-08-08 17:03:25.841081	192.168.21.193	192.168.14.250	HTTP	370	GET /ttest.html HTTP/1.1
5	2013-08-08 17:03:25.848466	192.168.14.250	192.168.21.193	TCP	1438	80 → 1055 [ACK] Seq=1015787007 Ack=1341544 Win=63928 Len=1380 [TCP segment of a reassembled PDU]
6	2013-08-08 17:03:25.848527	192.168.14.250	192.168.21.193	HTTP	698	HTTP/1.1 404 Not Found (text/html)
7	2013-08-08 17:03:25.858445	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1341544 Ack=1015789027 Win=65535 Len=0
8	2013-08-08 17:03:34.391749	192.168.21.193	192.168.14.250	HTTP	369	GET /test.html HTTP/1.1
9	2013-08-08 17:03:34.395487	192.168.14.250	192.168.21.193	HTTP	586	HTTP/1.1 200 OK (text/html)
10	2013-08-08 17:03:34.606352	192.168.21.193	192.168.14.250	TCP	58	1055 → 80 [ACK] Seq=1341855 Ack=1015789555 Win=65007 Len=0
11	2013-08-08 17:03:40.739601	192.168.21.193	192.168.14.250	HTTP	483	GET /test.html HTTP/1.1
12	2013-08-08 17:03:40.741538	192.168.14.250	192.168.21.193	HTTP	271	HTTP/1.1 304 Not Modified

Esta imagem mostra a captura realizada na interface NGFW OUTSIDE.



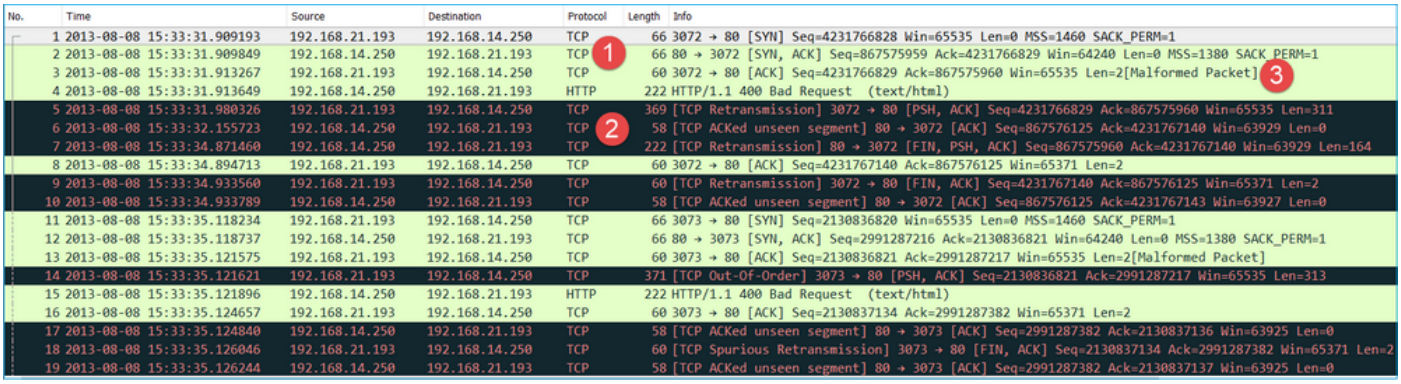
No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 17:03:25.554872	192.168.21.193	192.168.14.250	TCP	66	1055 → 80 [SYN] Seq=1839800324 Win=65535 Len=0 MSS=1380 SACK_PERM=1
2	2013-08-08 17:03:25.555177	192.168.14.250	192.168.21.193	TCP	66	80 → 1055 [SYN, ACK] Seq=521188628 Ack=1839800325 Win=64240 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

Pontos principais:

1. As duas capturas são quase idênticas (considere a aleatorização ISN).
2. Não há indicações de perda de pacotes.
3. Nenhum pacote fora de serviço (OOO)
4. Há 3 solicitações HTTP GET. O primeiro recebe uma mensagem de redirecionamento 404 ‘Não encontrado’, o segundo recebe uma mensagem de redirecionamento 200 ‘OK’ e o terceiro recebe uma mensagem de redirecionamento 304 ‘Não modificado’.

Capturas - Cenário de falha conhecida:

O conteúdo da captura de entrada (CAPI).

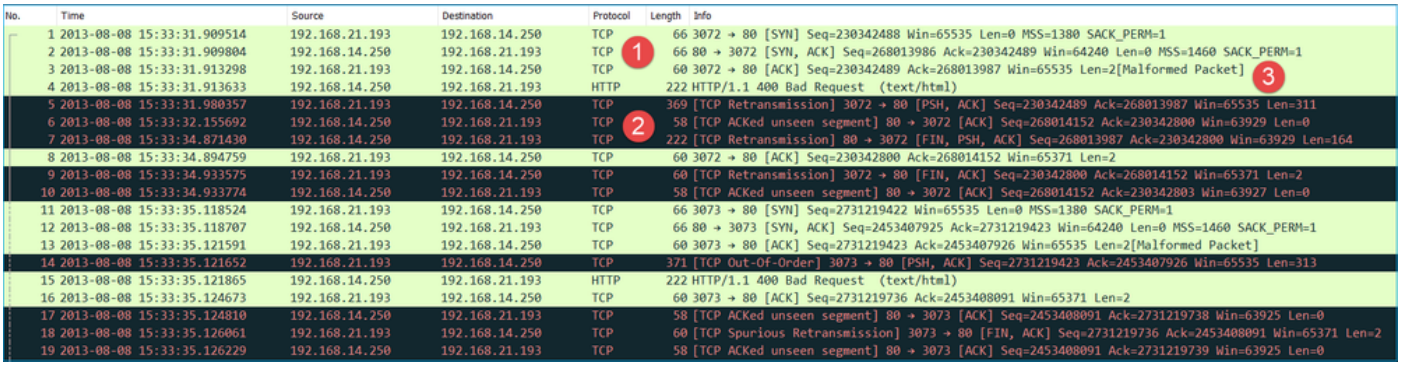


No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909193	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	2013-08-08 15:33:31.909849	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=867575959 Ack=4231766829 Win=64240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

Pontos principais:

1. Há um handshake triplo do TCP.
2. Há retransmissões de TCP e indicações de perda de pacotes.
3. Há um pacote (TCP ACK) que é identificado pelo Wireshark como Malformado.

Esta imagem mostra o conteúdo da captura de saída (CAPO).



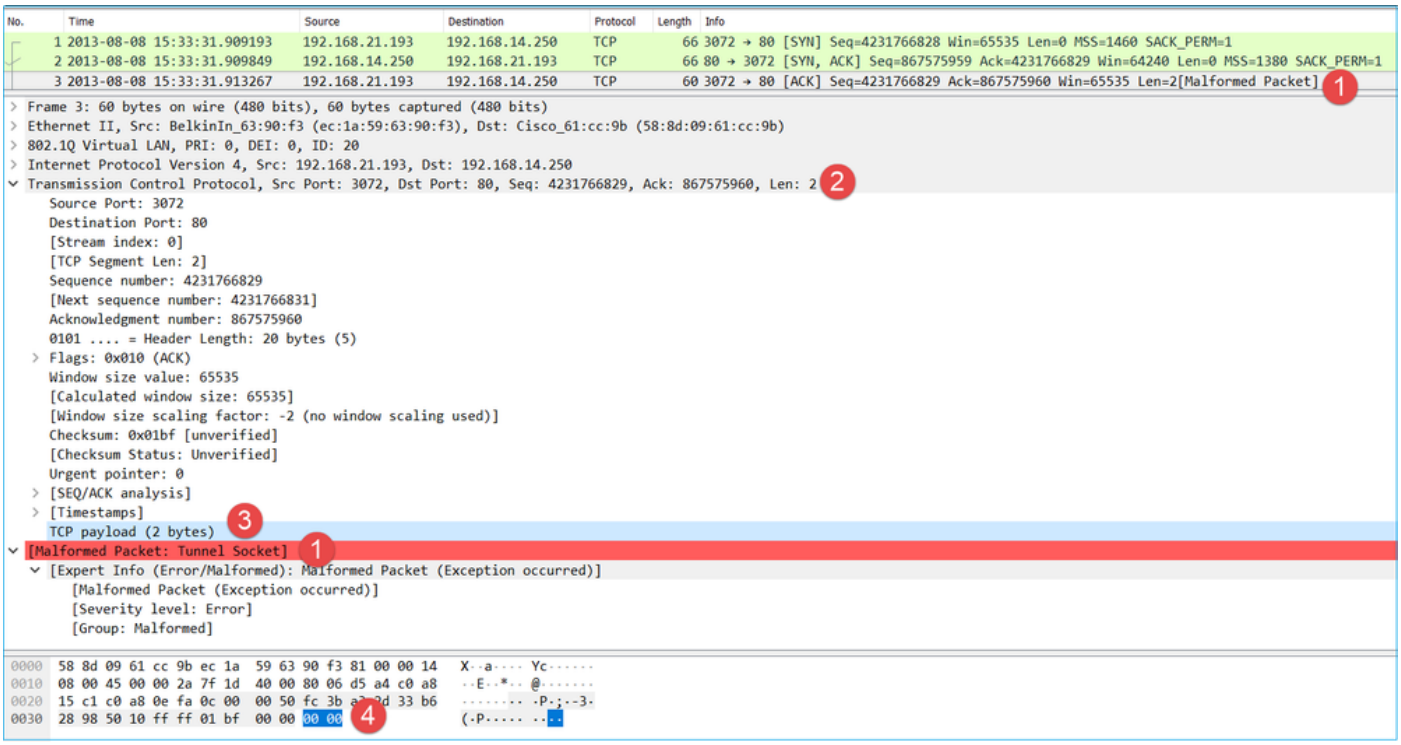
No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909514	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=230342488 Win=65535 Len=0 MSS=1380 SACK_PERM=1
2	2013-08-08 15:33:31.909804	192.168.14.250	192.168.21.193	TCP	66	80 → 3072 [SYN, ACK] Seq=268013986 Ack=230342489 Win=64240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

Pontos principais:

As duas capturas são quase idênticas (considere a aleatorização ISN):

1. Há um handshake triplo do TCP.
2. Há retransmissões de TCP e indicações de perda de pacotes.
3. Há um pacote (TCP ACK) que é identificado pelo Wireshark como Malformado.

Verifique o pacote malformado:



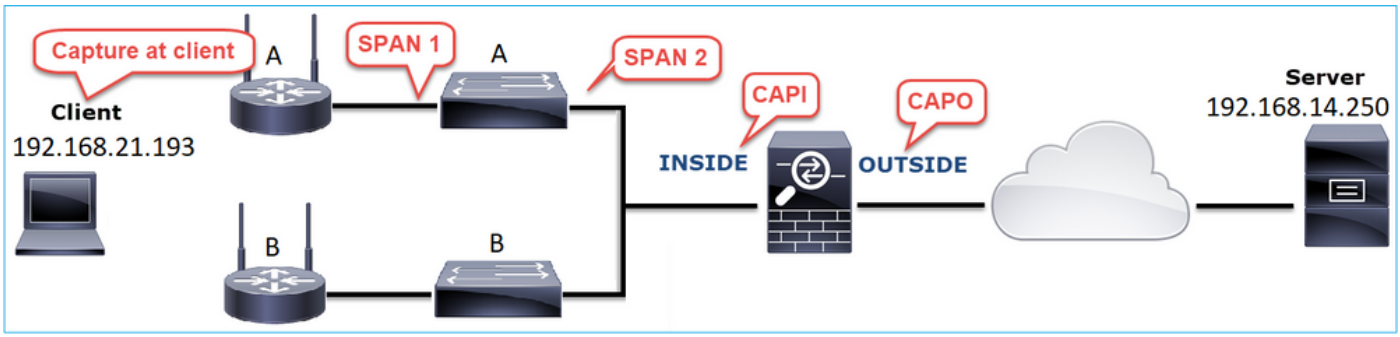
Pontos principais:

1. O pacote é identificado como Malformado pelo Wireshark.
2. Ele tem um comprimento de 2 bytes.
3. Há um payload TCP de 2 bytes.
4. O payload é de 4 zeros extras (00 00).

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Faça capturas adicionais. Inclua capturas nos endpoints e, se possível, tente aplicar o método divide and conquer para isolar a origem da corrupção de pacotes, por exemplo:

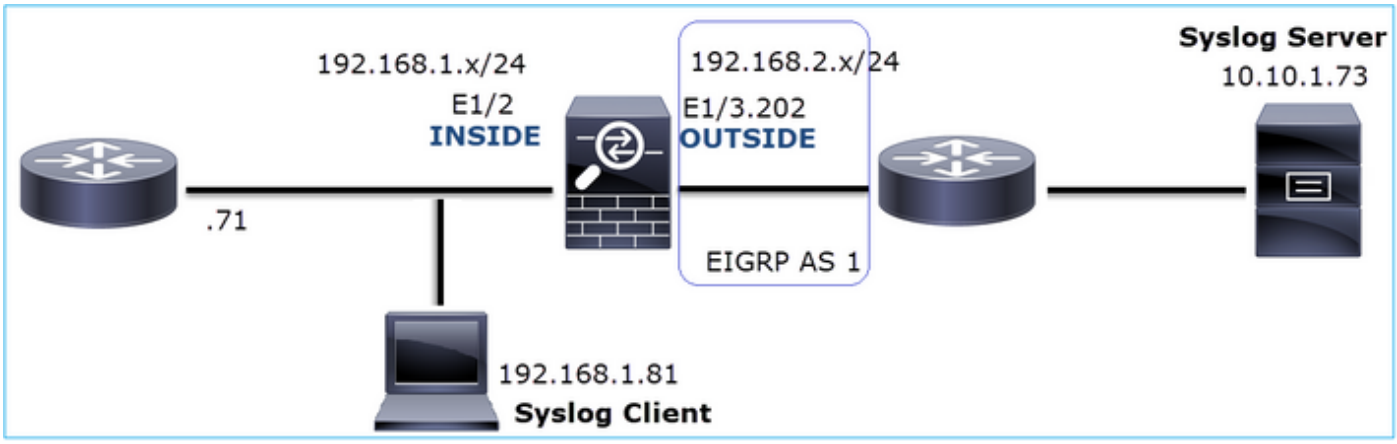


Nesse caso, os 2 bytes extras foram adicionados pelo driver de interface do switch 'A' e a solução foi substituir o switch que causa a corrupção.

Caso 8. Problema de conectividade UDP (pacotes ausentes)

Descrição do problema: as mensagens Syslog (UDP 514) não são vistas no Servidor Syslog de destino.

Esta imagem mostra a topologia:



Fluxo afetado:

IP orig.: 192.168.1.81

IP do Horário de Verão: 10.10.1.73

Protocolo: UDP 514

Capturar análise

Habilitar capturas no mecanismo 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
```

As capturas de FTD não mostram pacotes:

```
<#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
```

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Verifique a tabela de conexão do FTD.

Para verificar uma conexão específica, use esta sintaxe:

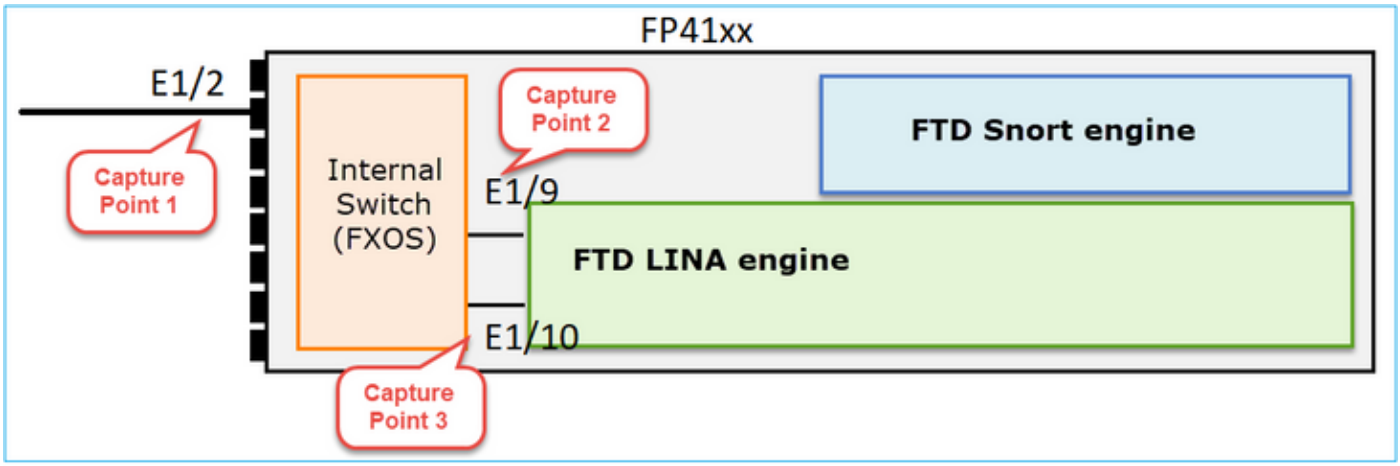
```
<#root>
firepower#
show conn address 192.168.1.81 port 514

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
```

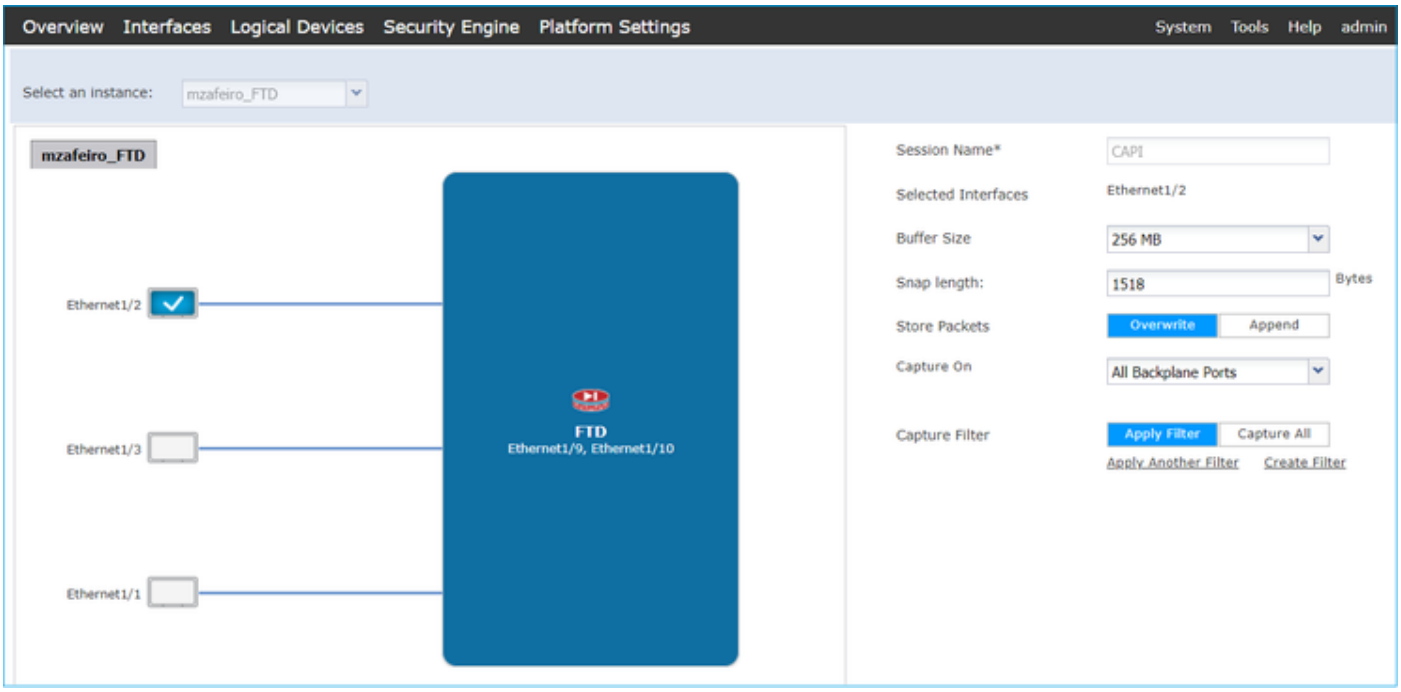
Pontos principais:

1. As interfaces de entrada e saída são as mesmas (curva-U).
2. O número de bytes tem um valor significativamente grande (~5 GBytes).
3. A marca "o" indica o descarregamento em fluxo (fluxo acelerado HW). Essa é a razão pela qual as capturas de FTD não mostram nenhum pacote. O descarregamento de fluxo é suportado apenas nas plataformas 41xx e 93xx. Nesse caso, o dispositivo é um 41xx.

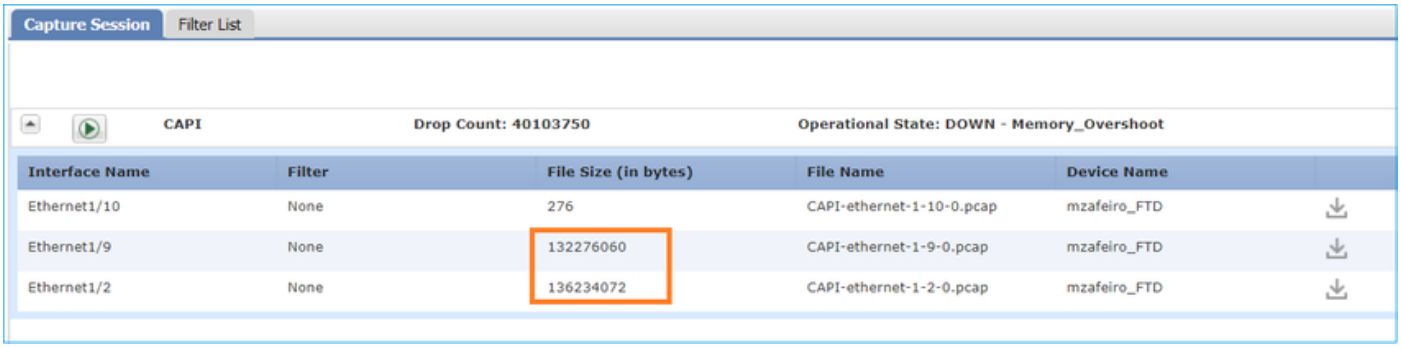
Ação 2. Faça capturas no nível do chassi.

Conecte-se ao gerenciador de chassi do Firepower e habilite a captura na interface de entrada (E1/2 nesse caso) e nas interfaces do backplane (E1/9 e E1/10), conforme mostrado na imagem:



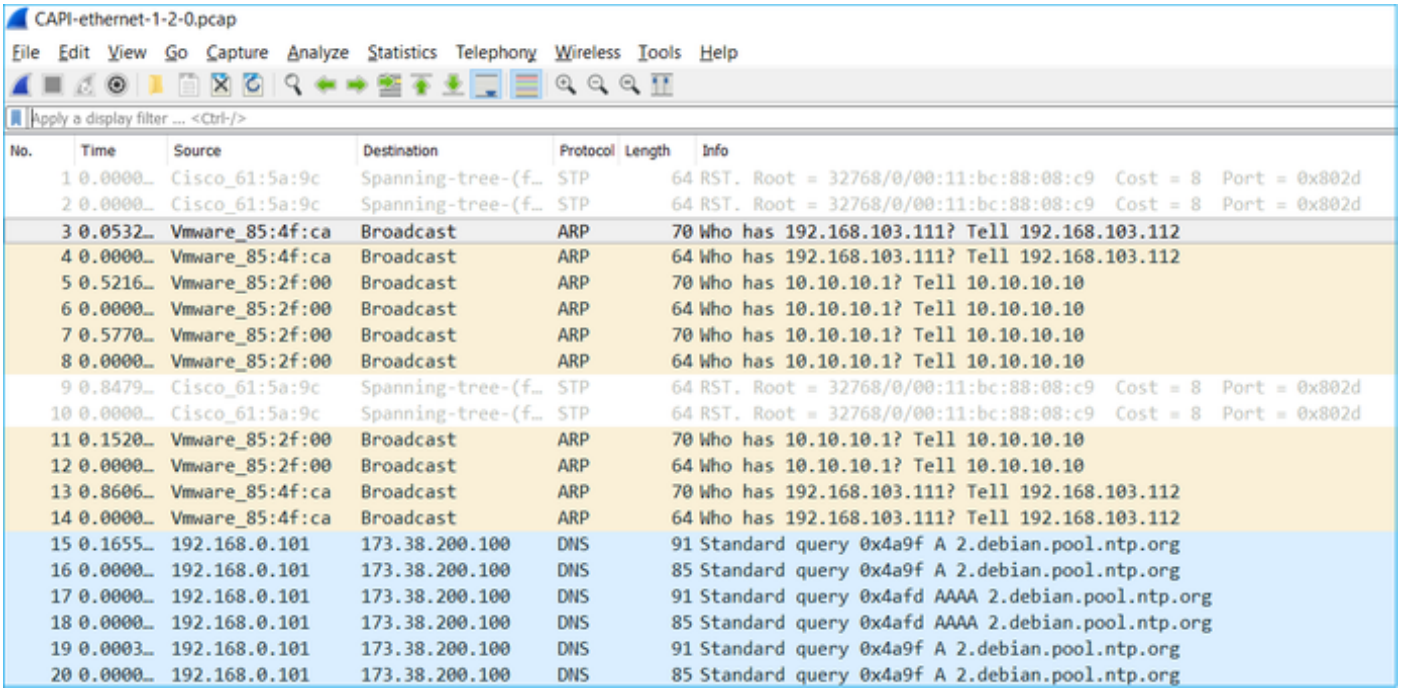


Depois de alguns segundos:

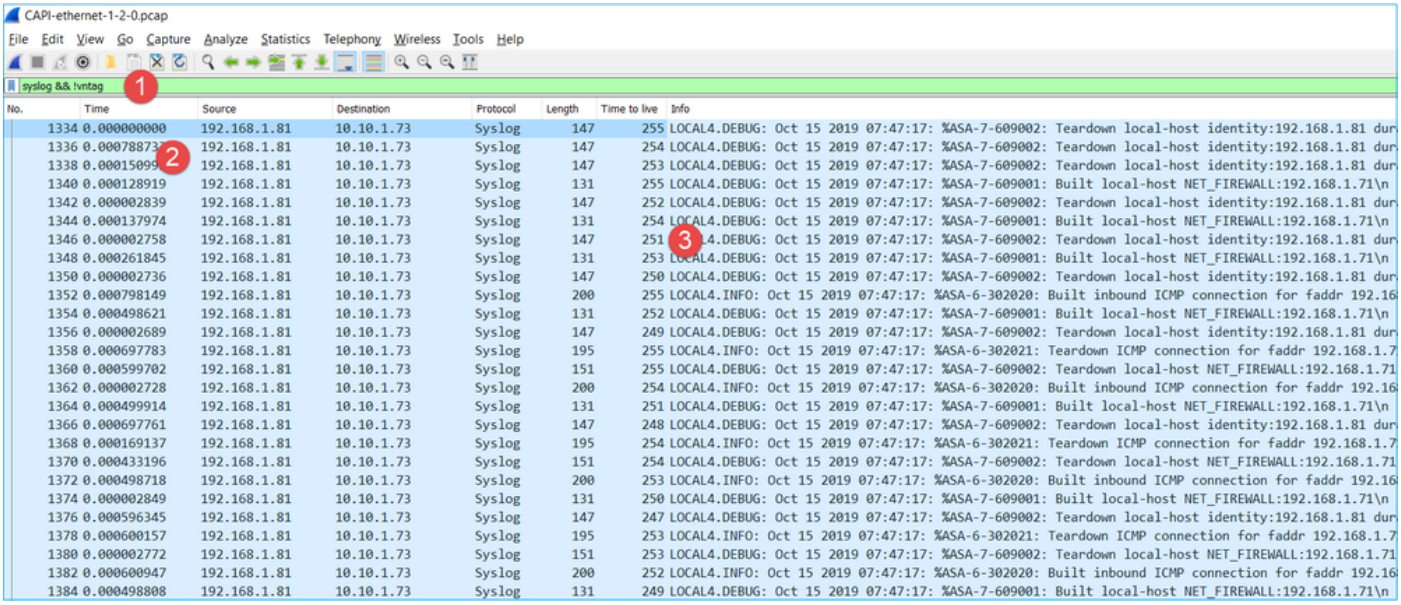


Dica: no Wireshark, exclua os pacotes marcados com VN para eliminar a duplicação de pacotes no nível da interface física

Antes:

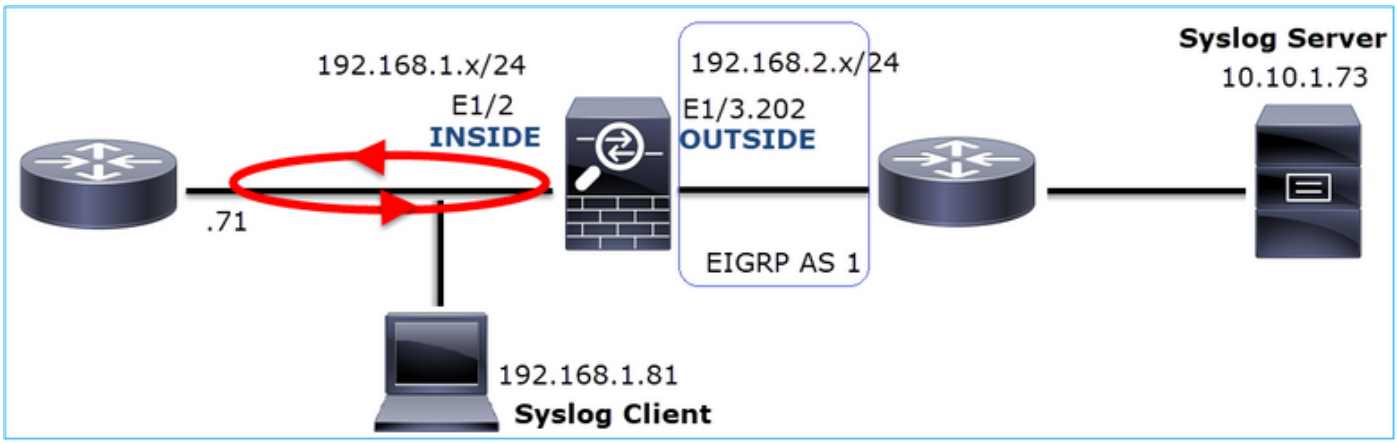


Após:



Pontos principais:

1. Um filtro de exibição é aplicado para remover duplicatas de pacotes e mostrar apenas syslogs.
2. A diferença entre os pacotes está no nível de microssegundos. Isso indica uma taxa de pacotes muito alta.
3. O valor do Time to Live (TTL) diminui continuamente. Isso indica um loop de pacote.



Ação 3. Use o packet-tracer.

Como os pacotes não passam pelo mecanismo LINA do firewall, você não pode fazer um rastreamento ao vivo (captura com rastreamento), mas pode rastrear um pacote emulado com o 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
```

Ação 4. Confirme o roteamento FTD.

Verifique a tabela de roteamento do firewall para ver se há algum problema de roteamento:

```
<#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
```

Pontos principais:

1. A rota aponta para a interface de saída correta.
2. A rota foi aprendida há alguns minutos (0:02:37).

Ação 5. Confirme a disponibilidade da conexão.

Verifique o tempo de atividade da conexão para ver quando esta conexão foi estabelecida:

```
<#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
```

Ponto-chave:

1. A conexão foi estabelecida há aproximadamente 4 minutos (isso ocorre antes da instalação da rota EIGRP na tabela de roteamento)

Ação 6. Limpe a conexão estabelecida.

Nesse caso, os pacotes correspondem a uma conexão estabelecida e são roteados para uma interface de saída errada; isso causa um loop. Isso se deve à ordem de operações do firewall:

1. Pesquisa de conexão estabelecida (tem prioridade sobre a pesquisa da tabela de roteamento global).
2. Consulta de conversão de endereço de rede (NAT) - A fase UN-NAT (NAT de destino) tem precedência sobre a pesquisa de PBR e de rota.
3. Roteamento baseado em políticas (PBR)
4. Pesquisa de tabela de roteamento global

Como a conexão nunca atinge o tempo limite (o cliente Syslog envia continuamente pacotes enquanto o tempo limite de ociosidade de conexão UDP é de 2 minutos), é necessário limpar manualmente a conexão:

<#root>

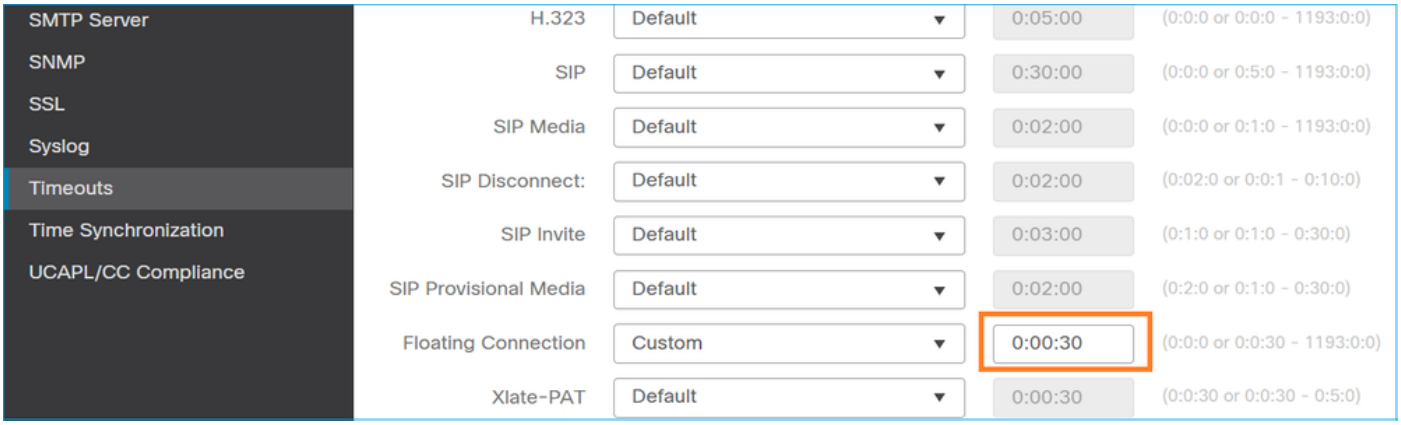
```
firepower#

clear conn address 10.10.1.73 address 192.168.1.81 protocol udp port 514

1 connection(s) deleted.
```

Verifique se uma nova conexão foi estabelecida:

```
<#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
```

Ação 7. Configure o tempo limite de conexão flutuante.

Essa é a solução adequada para resolver o problema e evitar o roteamento não ideal, especialmente para fluxos UDP. Navegue até Devices > Platform Settings > Timeouts e defina o valor:



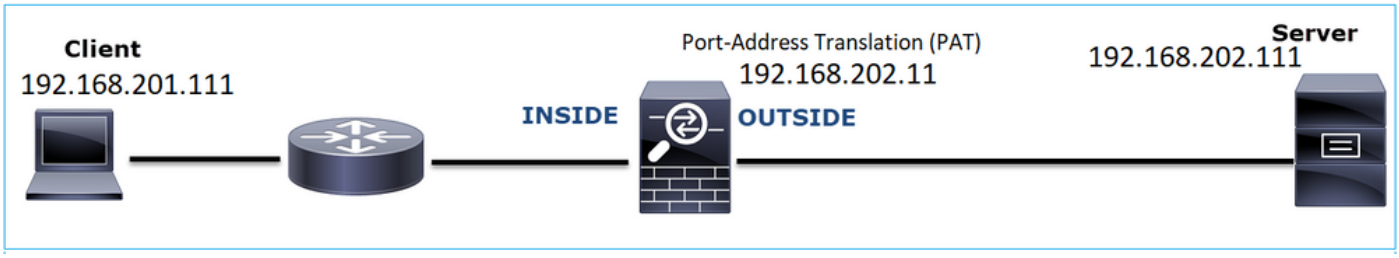
Você pode encontrar mais detalhes sobre o timeout de conexão flutuante na Referência de Comandos:

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 de conectividade HTTPS (Cenário 1)

Descrição do problema: a comunicação HTTPS entre o cliente 192.168.201.105 e o servidor 192.168.202.101 não pode ser estabelecida

Esta imagem mostra a topologia:



Fluxo afetado:

IP orig.: 192.168.201.111

IP do Horário de Verão: 192.168.202.111

Protocolo: TCP 443 (HTTPS)

Capturar análise

Habilitar capturas no mecanismo LINA FTD:

O IP usado na captura EXTERNA é diferente devido à configuração da conversão de endereço de porta.

<#root>

```
firepower#
capture CAPI int INSIDE match ip host 192.168.201.111 host 192.168.202.111
firepower#
capture CAPO int OUTSIDE match ip host 192.168.202.11 host 192.168.202.111
```

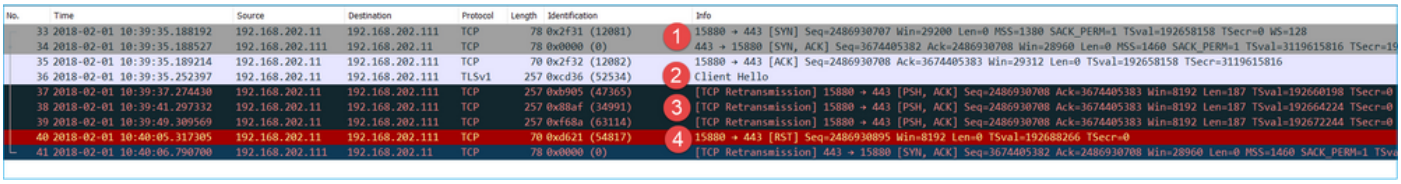
Esta imagem mostra a captura realizada na interface INSIDE do NGFW:

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)	1 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)	2 Client Hello
42	2018-02-01 10:39:35.252352	192.168.202.111	192.168.201.111	TCP	70	0xefb4 (61364)	3 443 → 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)	4 443 → 6666 [RST] Seq=4086514532 Win=8192 Len=0 TSval=3119645908 TSecr=0

Pontos principais:

1. Há um handshake triplo do TCP.
2. A negociação SSL é iniciada. O cliente envia uma mensagem Hello do cliente.
3. Há um TCP ACK enviado ao cliente.
4. Há um TCP RST enviado ao cliente.

Esta imagem mostra a captura realizada na interface NGFW OUTSIDE.



Pontos principais:

1. Há um handshake triplo do TCP.
2. A negociação SSL é iniciada. O cliente envia uma mensagem Hello do cliente.
3. Há retransmissões de TCP enviadas do firewall para o servidor.
4. Há um TCP RST enviado ao servidor.

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

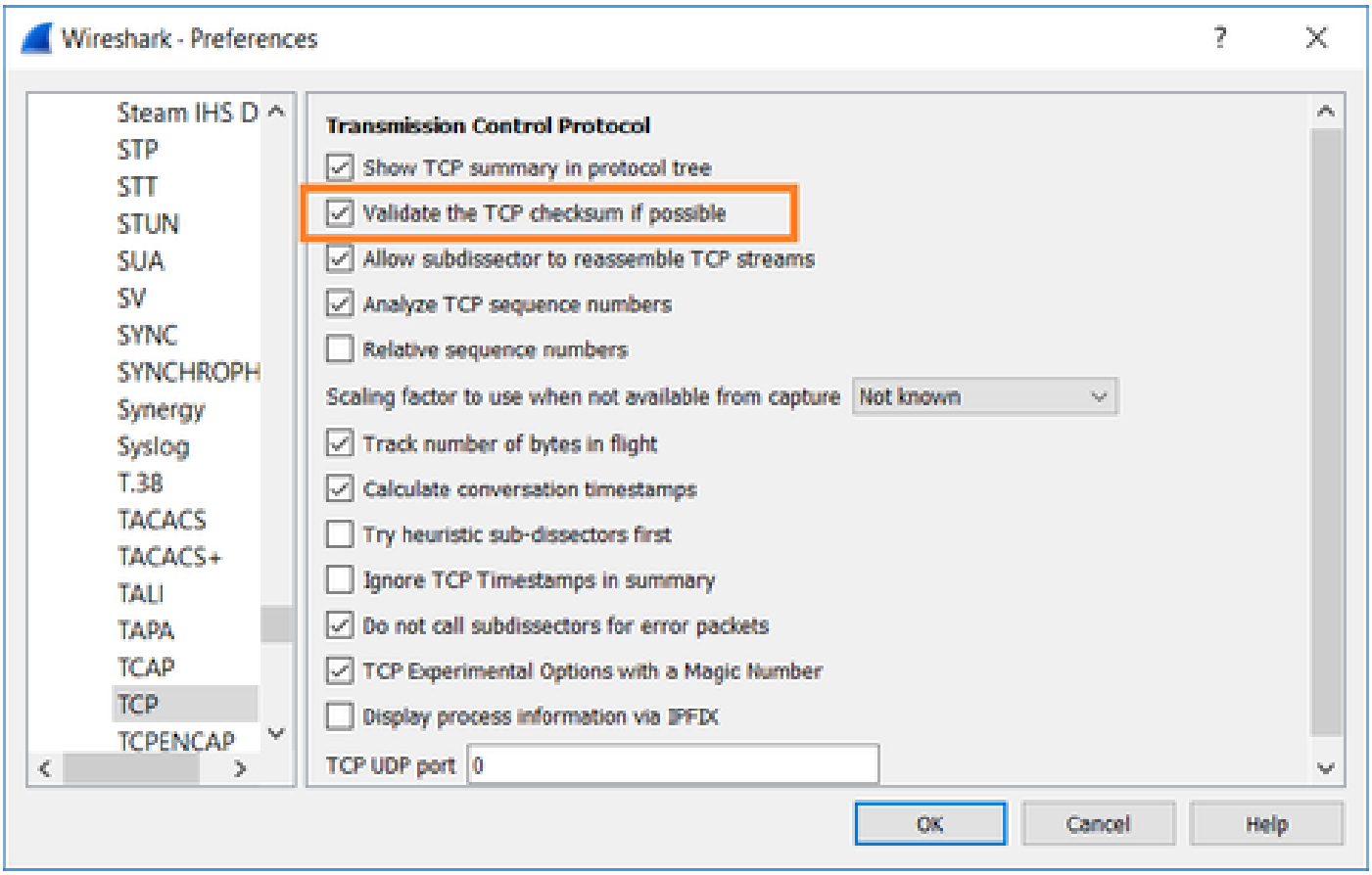
Ação 1. Faça capturas adicionais.

Uma captura feita no servidor revela que o servidor recebeu o Hellos do cliente TLS com checksum TCP corrompido e o descarta silenciosamente (não há TCP RST ou qualquer outro pacote de resposta para o cliente):

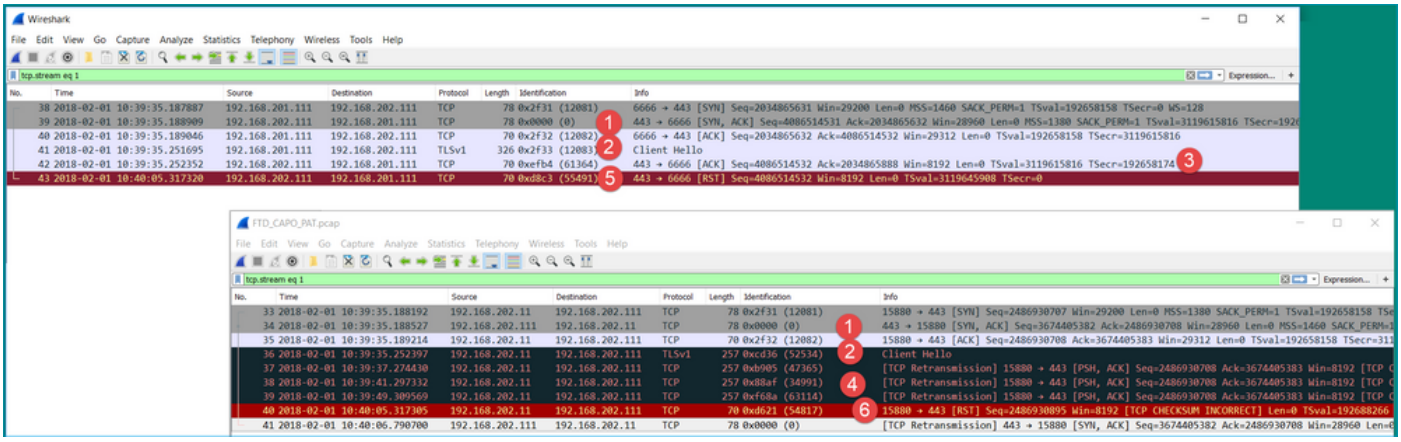
```
21:26:27.133677 IP (tos 0x0, ttl 64, id 52534, offset 0, flags [DF], proto TCP (6), length 239)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x0c65 (incorrect -> 0x3063), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192658174 ecr 3119615816], length 187
21:26:29.155652 IP (tos 0x0, ttl 64, id 47365, offset 0, flags [DF], proto TCP (6), length 239)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x4db7 (incorrect -> 0x71b5), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192660198 ecr 0], length 187
21:26:33.178142 IP (tos 0x0, ttl 64, id 34991, offset 0, flags [DF], proto TCP (6), length 239)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x3d[illegible]d (incorrect -> 0x61fb), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192664224 ecr 0], length 187
21:26:41.189640 IP (tos 0x0, ttl 64, id 63114, offset 0, flags [DF], proto TCP (6), length 239)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x1e[illegible]9 (incorrect -> 0x42a7), seq 1:188, ack 1, win 64, options [nop,nop,T
S val 192672244 ecr 0], length 187
21:26:57.195947 IP (tos 0x0, ttl 64, id 54817, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.202.11.15880 > 192.168.202.111.443: Flags [R], cksum 0x9ee[illegible] (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 você junta tudo:

Nesse caso, para entender, há uma necessidade de ativar no Wireshark a opção Validate the TCP checksum if possible. Navegue até Edit > Preferences > Protocols > TCP, conforme mostrado na imagem.

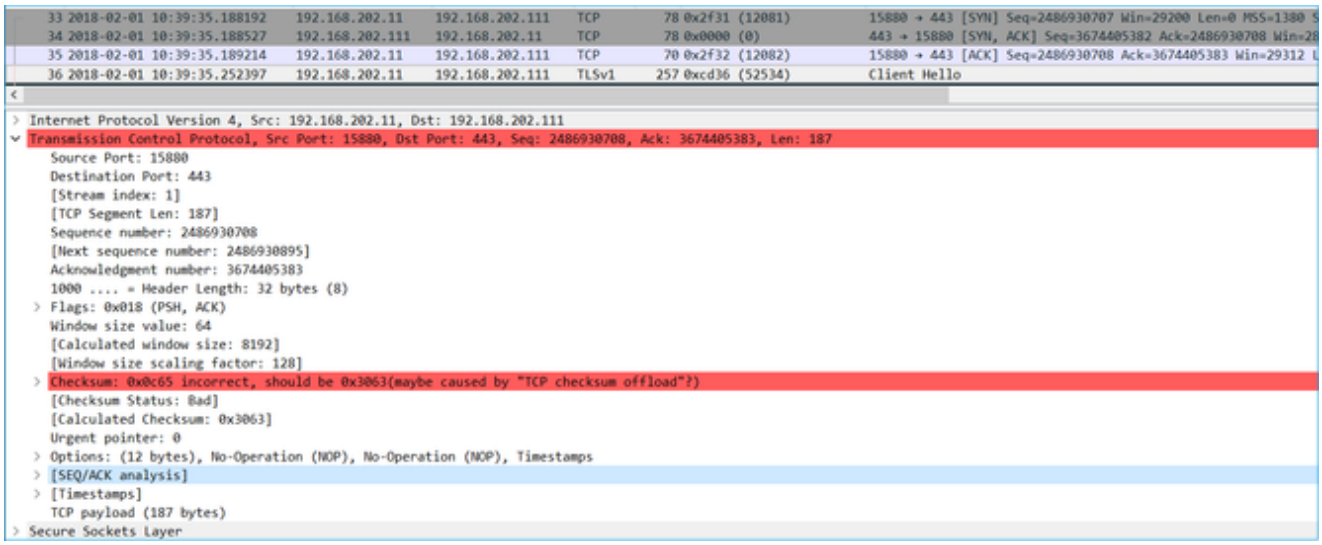


Nesse caso, é útil colocar as capturas lado a lado para obter a imagem completa:



Pontos principais:

1. Há um handshake triplo do TCP. As IDs de IP são as mesmas. Isso significa que o fluxo não foi intermediado por proxy pelo firewall.
2. Um Hello do cliente TLS vem do cliente com o ID IP 12083. O pacote recebe proxy do firewall (o firewall, nesse caso, foi configurado com a Política de descriptografia TLS) e a ID IP é alterada para 52534. Além disso, a soma de verificação TCP do pacote é corrompida (devido a um defeito de software que depois foi corrigido).
3. O firewall está no modo Proxy TCP e envia um ACK ao cliente (que falsifica o servidor).



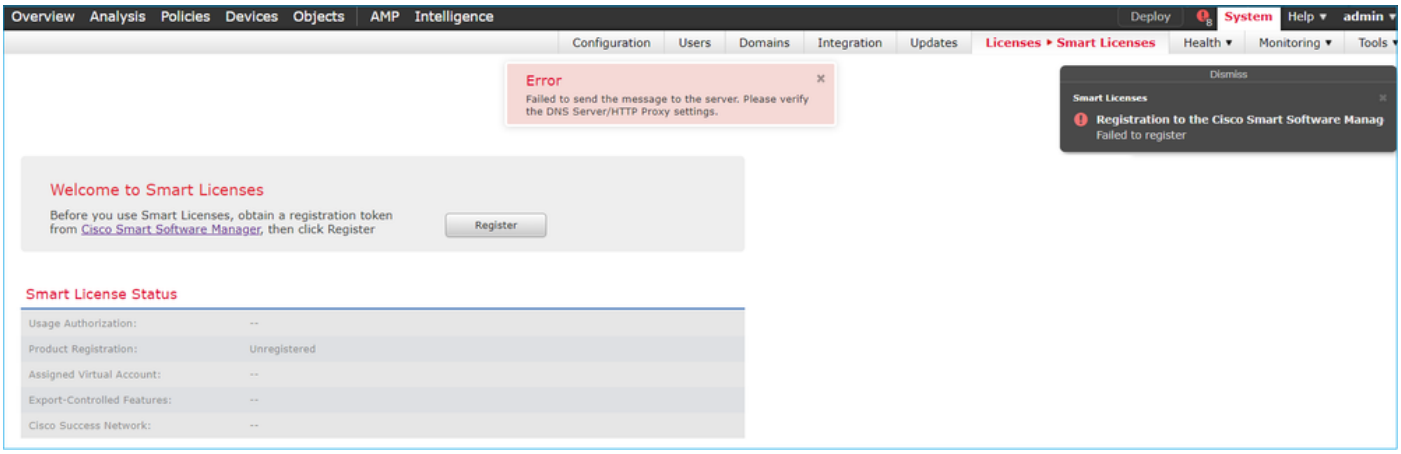
4. O firewall não recebe nenhum pacote TCP ACK do servidor e retransmite a mensagem Hello do cliente TLS. Isso ocorre novamente devido ao modo Proxy TCP ativado pelo firewall.
5. Após ~30 segundos, o firewall desiste e envia um TCP RST para o cliente.
6. O firewall envia um TCP RST para o servidor.

Referência:

[Processamento de handshake TLS/SSL do Firepower]

Caso 10. Problema de conectividade HTTPS (Cenário 2)

Descrição do problema: Falha no registro da licença inteligente do FMC.



Esta imagem mostra a topologia:

Fluxo afetado:

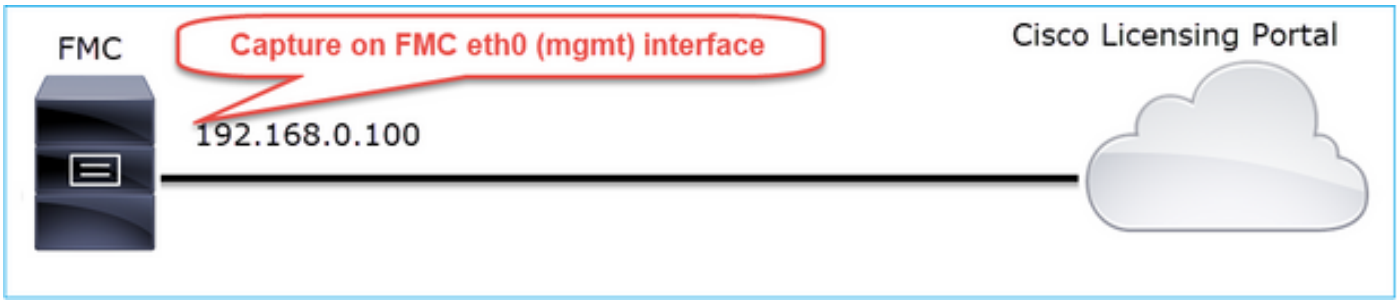
IP orig.: 192.168.0.100

Dst: tools.cisco.com

Protocolo: TCP 443 (HTTPS)

Capturar análise

Permitir a captura na interface de gestão do CVP:

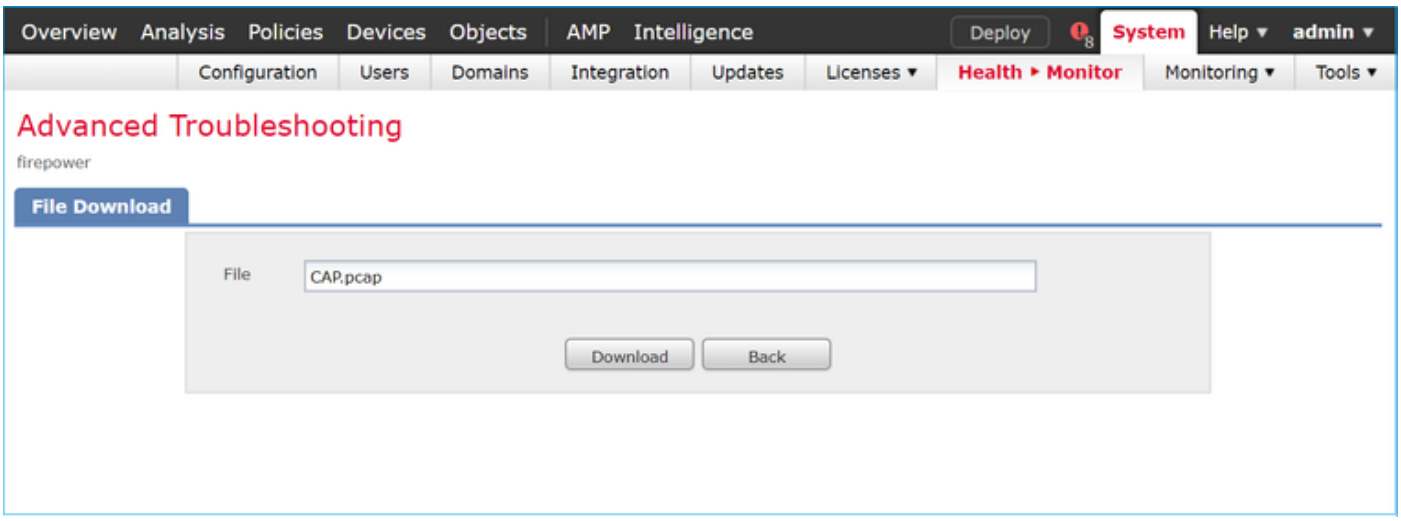


Tente se registrar novamente. Quando a mensagem de erro for exibida, pressione CTRL-C para interromper a captura:

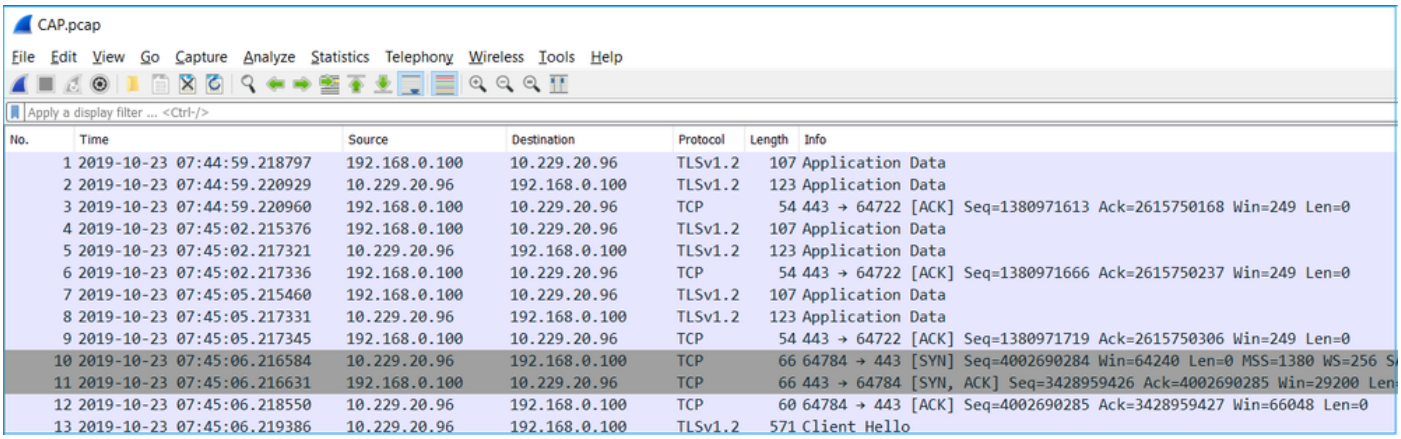
<#root>

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

tcpdump -i eth0 port 443 -s 0 -w CAP.pcap

HS_PACKET_BUFFER_SIZE is set to 4.
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

^C

264 packets captured

<- CTRL-C

264 packets received by filter
0 packets dropped by kernel

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

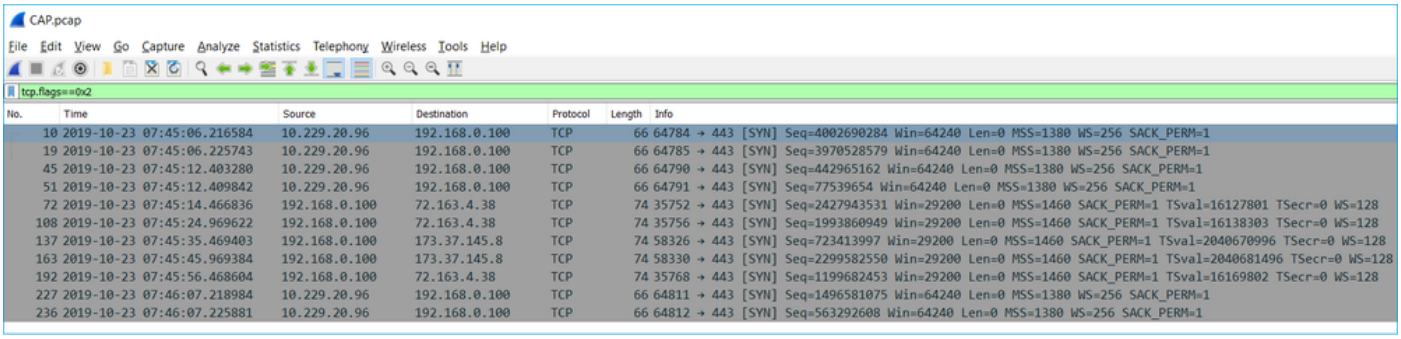
Colete a captura do FMC (System > Health > Monitor, selecione o dispositivo e selecione Advanced Troubleshooting), como mostrado na imagem:



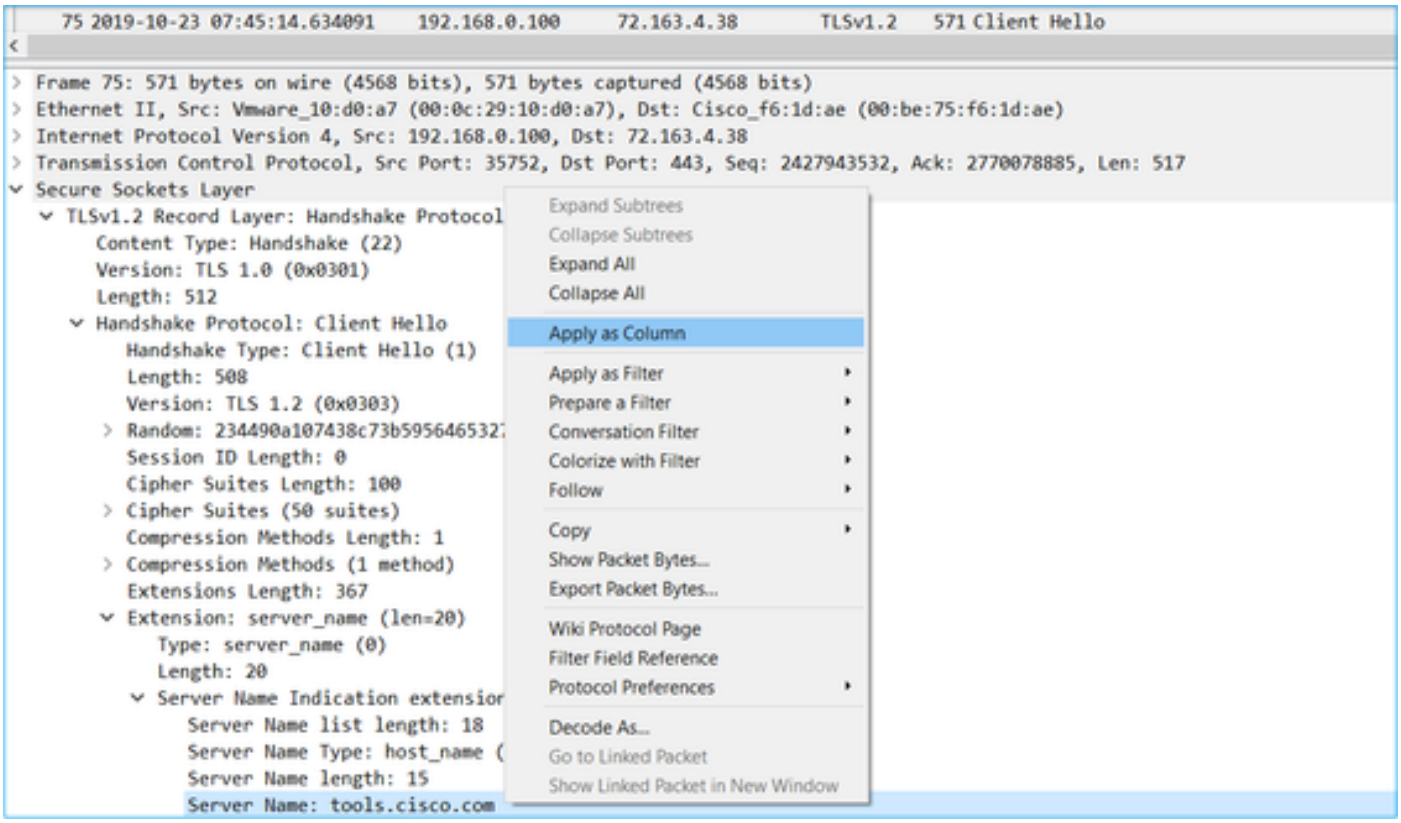
A imagem mostra a captura FMC no Wireshark:



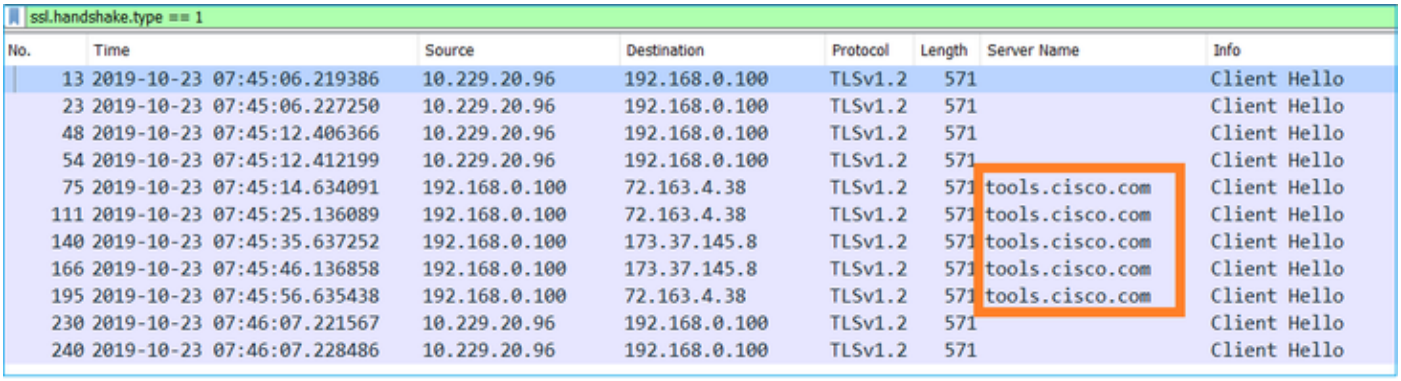
Dica: para verificar todas as novas sessões TCP que foram capturadas, use o filtro de exibição tcp.flags==0x2 no Wireshark. Isso filtra todos os pacotes TCP SYN que foram capturados.



Dica: aplique como coluna o campo Server Name do Hello do cliente SSL.



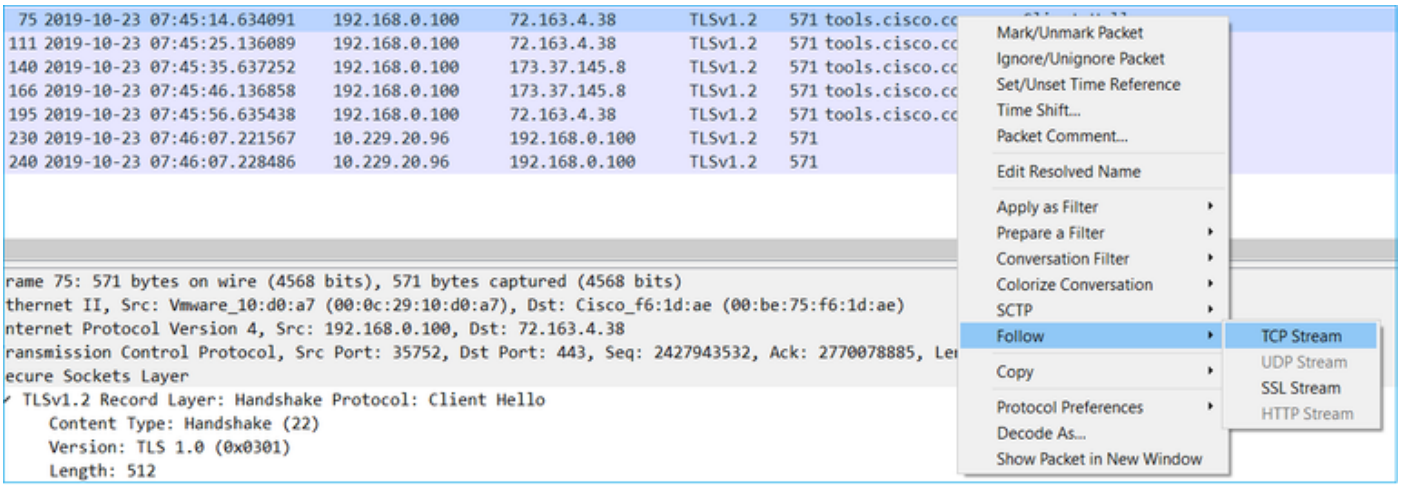
Dica: aplique este filtro de exibição para ver apenas as mensagens de Hello do cliente ssl.handshake.type == 1

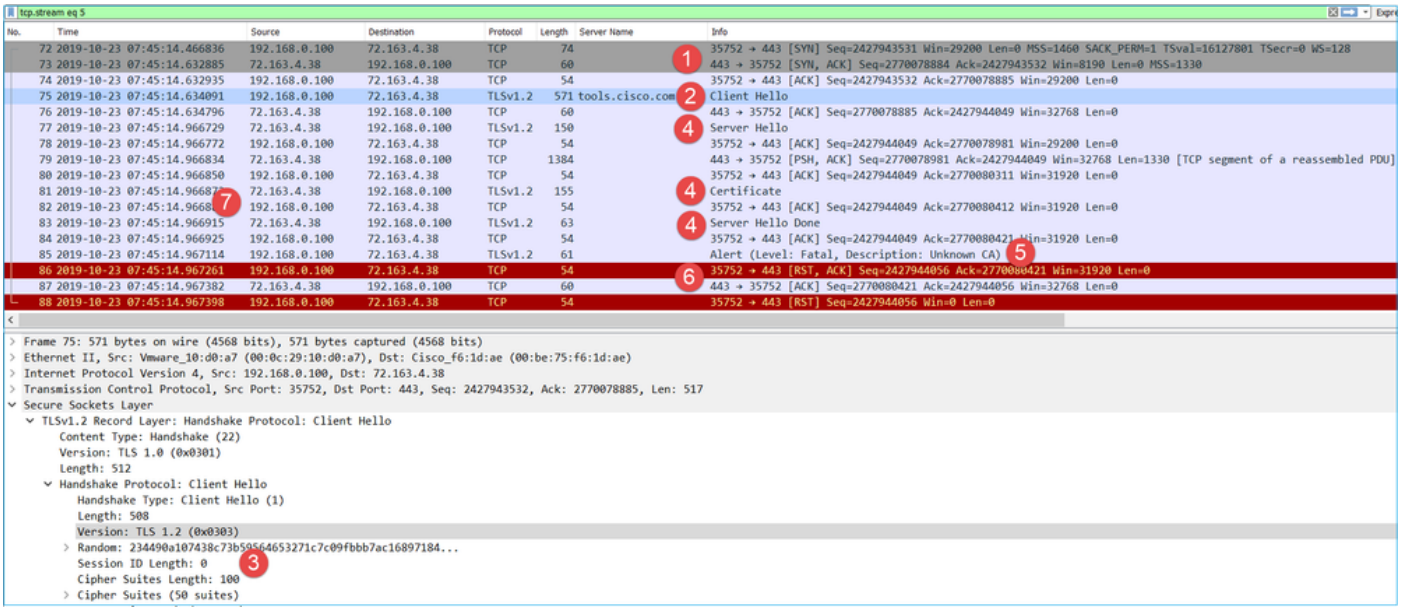


No.	Time	Source	Destination	Protocol	Length	Server Name	Info
13	2019-10-23 07:45:06.219386	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
23	2019-10-23 07:45:06.227250	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
48	2019-10-23 07:45:12.406366	10.229.20.96	192.168.0.100	TLSv1.2	571		Client 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

Observação: no momento em que este documento foi escrito, o portal Smart Licensing (tools.cisco.com) usa estes IPs: 72.163.4.38, 173.37.145.8

Siga um dos fluxos TCP (Follow > TCP Stream), como mostrado na imagem.

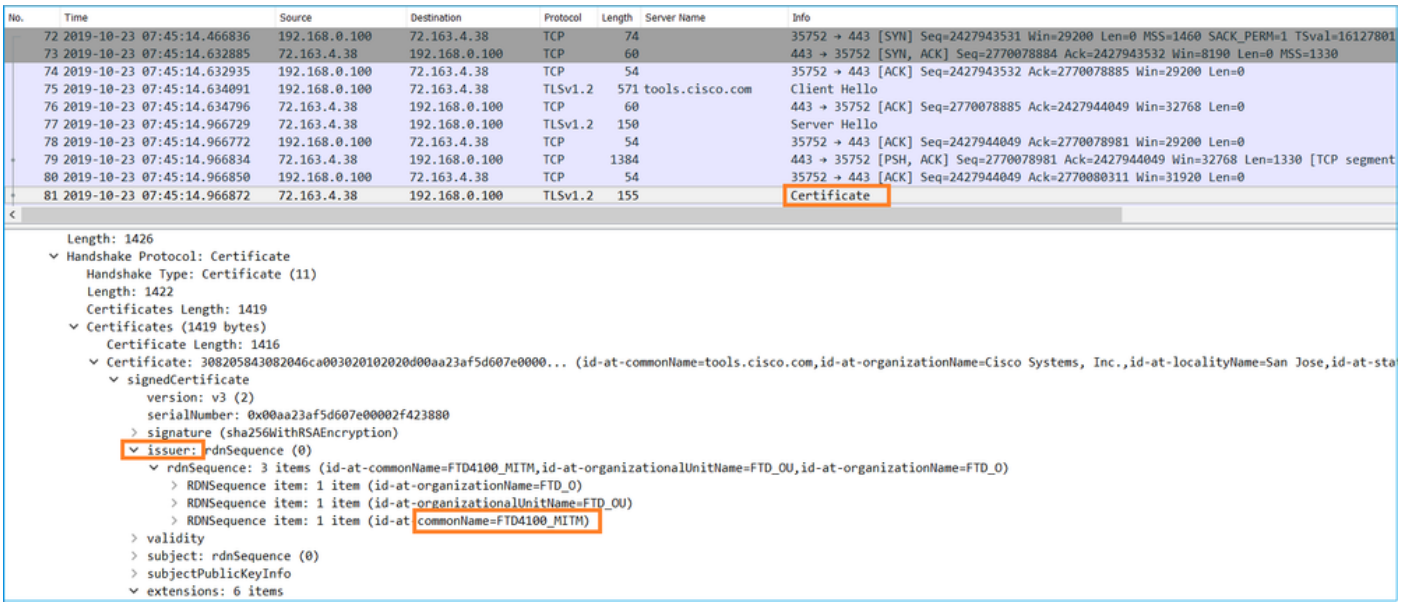




Pontos principais:

1. Há um handshake triplo do TCP.
2. O cliente (FMC) envia uma mensagem de saudação do cliente SSL para o portal Smart Licensing.
3. A ID da Sessão SSL é 0. Isso significa que não é uma sessão retomada.
4. O servidor de destino responde com a mensagem Hello do servidor, Certificado e Hello do servidor concluída.
5. O cliente envia um alerta fatal SSL referente a uma "CA desconhecida".
6. O cliente envia um TCP RST para fechar a sessão.
7. A duração total da sessão TCP (do estabelecimento ao fechamento) foi de aproximadamente 0,5 s.

Selecione o certificado do servidor e expanda o campo emissor para ver o commonName. Nesse caso, o nome comum revela um dispositivo que faz MITM (Man-in-the-middle).



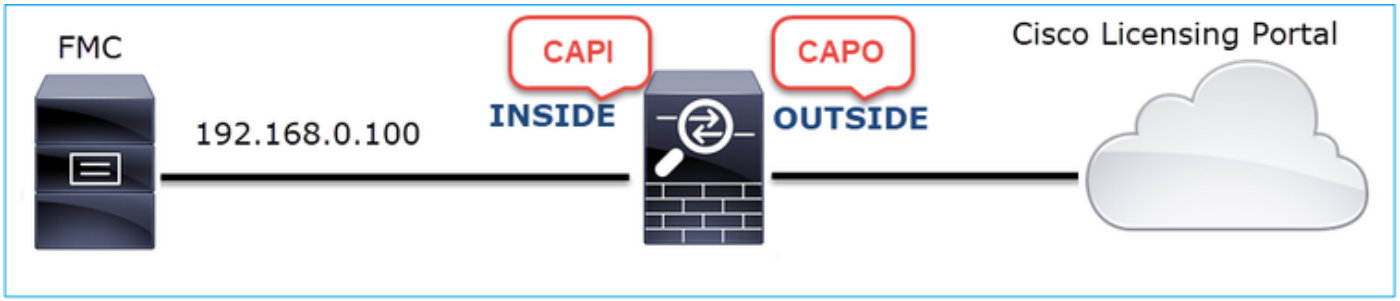
Isso é mostrado nesta imagem:

Ações recomendadas

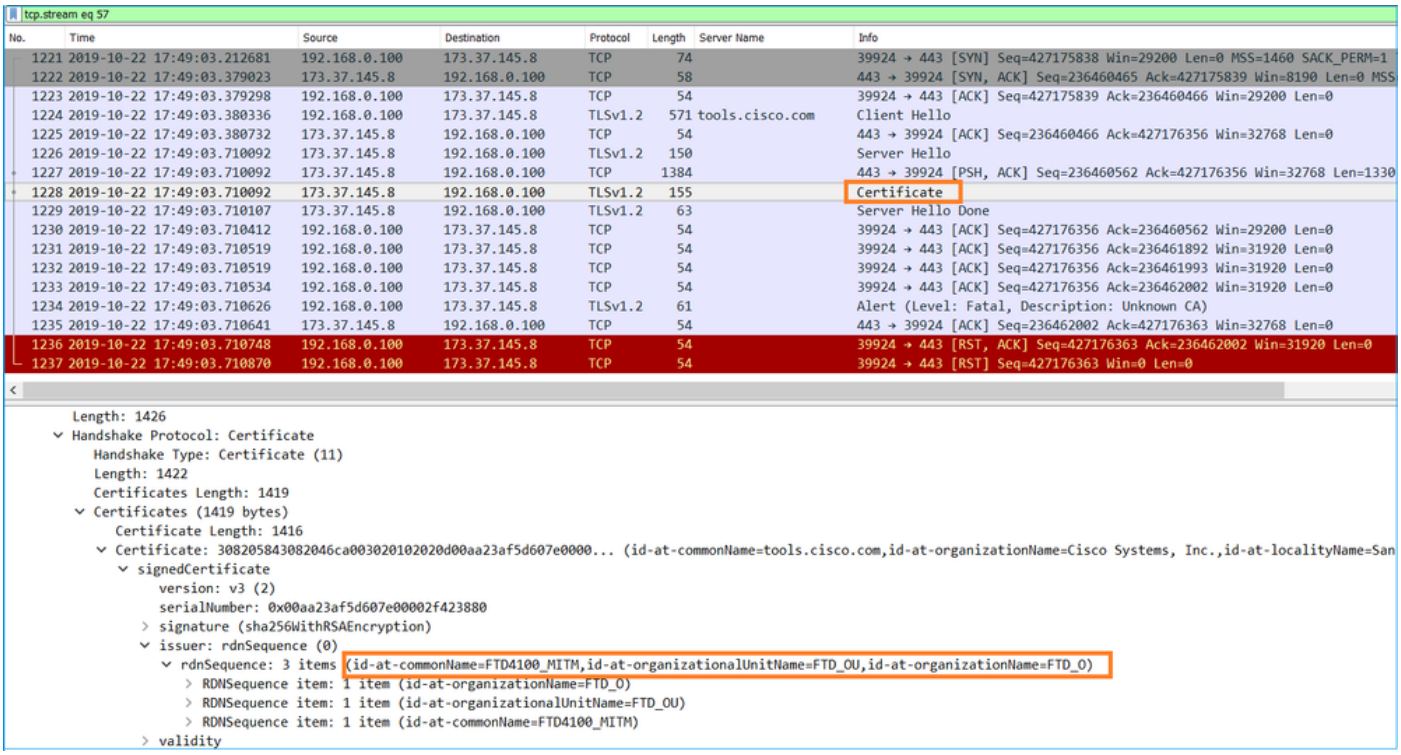
As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Faça capturas adicionais.

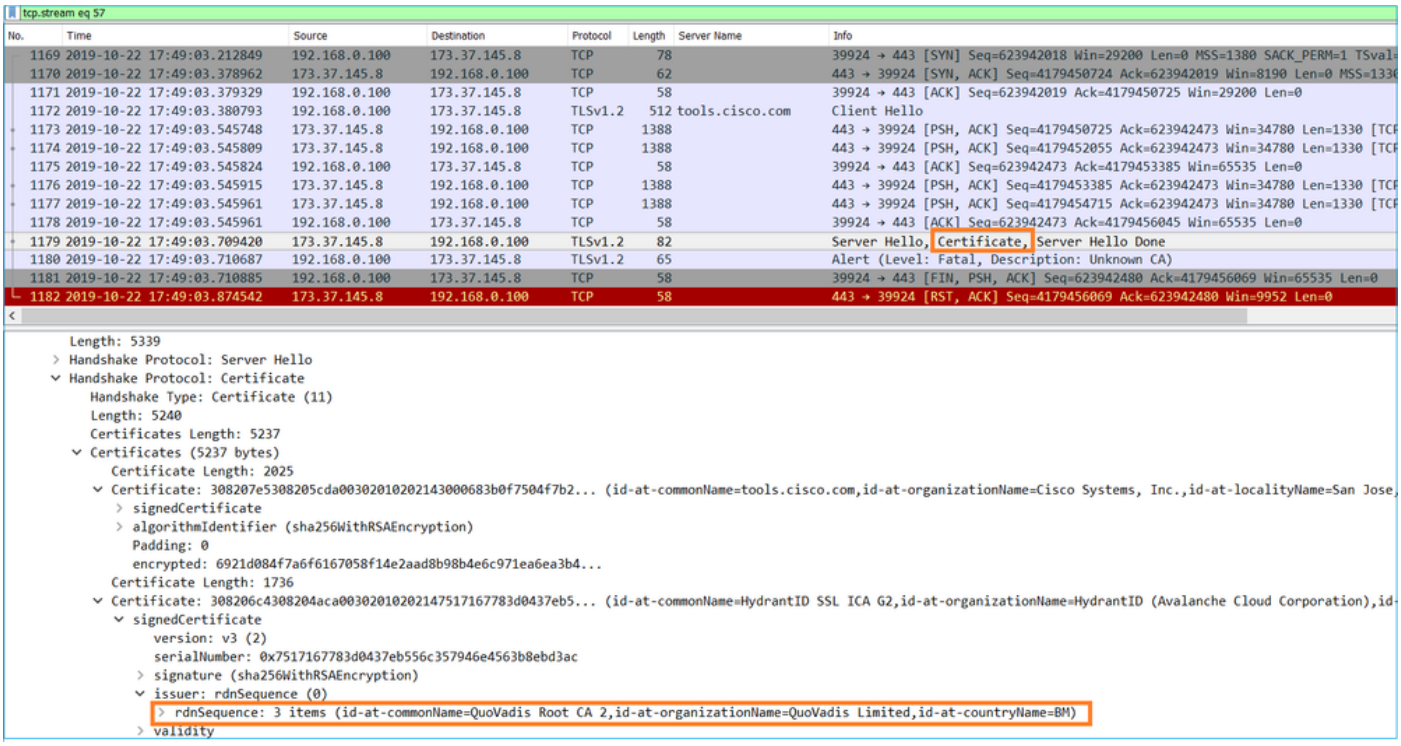
Faça capturas no dispositivo de firewall de trânsito:



A CAPI mostra:



O CAPO mostra:



Essas capturas comprovam que o firewall de trânsito modifica o certificado do servidor (MITM)

Ação 2. Verifique os logs do dispositivo.

Você pode coletar o pacote FMC TS conforme descrito neste documento:

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

Nesse caso, o arquivo /dir-archives/var-log/process_stdout.log mostra mensagens como esta:

<#root>

```
SOUT: 10-23 05:45:14 2019-10-23 05:45:36 sla[10068]: *Wed .967 UTC: CH-LIB-ERROR: ch_pf_curl_send_msg[4
failed to perform, err code 60, err string "SSL peer certificate or SSH remote key was not OK"
...
SOUT: 10-23 05:45:14 2019-10-23 05:45:36 sla[10068]: *Wed .967 UTC: CH-LIB-TRACE: ch_pf_curl_is_cert_is
cert issue checking, ret 60, url "https://tools.cisco.com/its/
```

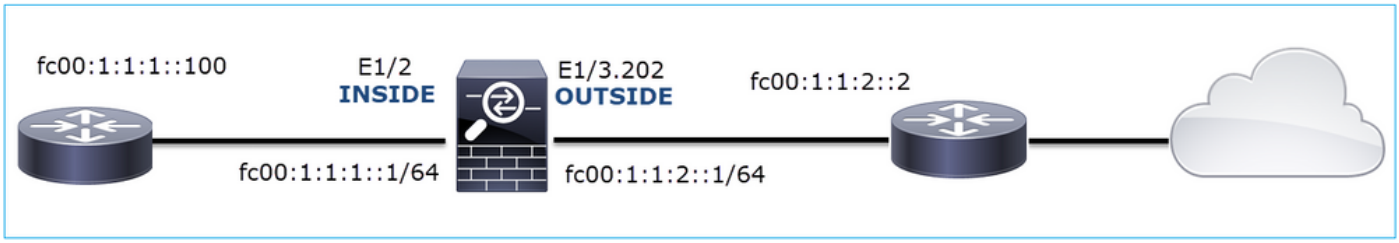
Solução recomendada

Desabilite o MITM para o fluxo específico para que o FMC possa se registrar com êxito na nuvem do Smart Licensing.

Caso 11. Problema de conectividade IPv6

Descrição do problema: os hosts internos (localizados atrás da interface INTERNA do firewall) não podem se comunicar com os hosts externos (hosts localizados atrás da interface EXTERNA do firewall).

Esta imagem mostra a topologia:



Fluxo afetado:

IP orig.: fc00:1:1:1::100

IP do Horário de Verão: fc00:1:1:2::2

Protocolo: qualquer

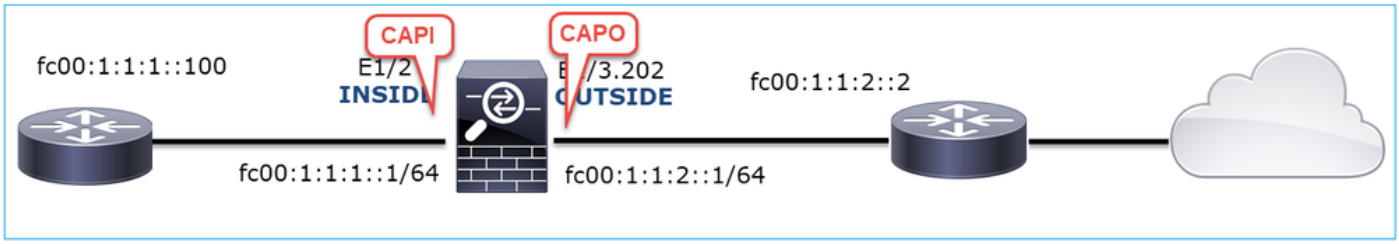
Capturar análise

Habilitar capturas no mecanismo LINA FTD.

<#root>

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

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



Capturas - Cenário não funcional

Essas capturas foram feitas em paralelo com um teste de conectividade ICMP de IP fc00:1:1:1::100 (roteador interno) para IP fc00:1:1:2::2 (roteador upstream).

A captura na interface INSIDE do firewall contém:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 13:02:07.001663	fc00:1:1:1::100	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
2	2019-10-24 13:02:07.001876	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6	86	Neighbor Advertisement fc00:1:1:1::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:ae
3	2019-10-24 13:02:07.002273	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=0, hop limit=64 (no response found!)
4	2019-10-24 13:02:08.997918	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
5	2019-10-24 13:02:10.998056	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
6	2019-10-24 13:02:11.999917	fe80::2be:75ff:fef6:1dae	fc00:1:1:1::100	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::100 from 00:be:75:f6:1d:ae
7	2019-10-24 13:02:12.002075	fc00:1:1:1::100	fe80::2be:75ff:fef6:1dae	ICMPv6	78	Neighbor Advertisement fc00:1:1:1::100 (rtr, sol)
8	2019-10-24 13:02:12.998346	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
9	2019-10-24 13:02:14.998483	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
10	2019-10-24 13:02:17.062725	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1dae	ICMPv6	86	Neighbor Solicitation for fe80::2be:75ff:fef6:1dae from 4c:4e:35:fc:fc:d8
11	2019-10-24 13:02:17.062862	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	78	Neighbor Advertisement fe80::2be:75ff:fef6:1dae (rtr, sol)
12	2019-10-24 13:02:22.059994	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	86	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8 from 00:be:75:f6:1d:ae
13	2019-10-24 13:02:22.063000	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1dae	ICMPv6	78	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

Pontos principais:

1. O roteador envia uma mensagem de solicitação de vizinho IPv6 e solicita o endereço MAC do dispositivo upstream (IP fc00:1:1:1::1).
2. O firewall responde com um anúncio de vizinho IPv6.
3. O roteador envia uma solicitação de eco ICMP.
4. O firewall envia uma mensagem de solicitação de vizinho IPv6 e solicita o endereço MAC do dispositivo downstream (fc00:1:1:1::100).
5. O roteador responde com um anúncio de vizinho IPv6.
6. O roteador envia solicitações adicionais de eco ICMP IPv6.

A captura na interface EXTERNA do firewall contém:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 13:02:07.002517	fe80::2be:75ff:fef6:1d8e	ff02::1:ff00:2	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
2	2019-10-24 13:02:07.005569	fc00:1:1:2::2	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
3	2019-10-24 13:02:08.997995	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
4	2019-10-24 13:02:09.001815	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
5	2019-10-24 13:02:10.025938	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
6	2019-10-24 13:02:10.998132	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
7	2019-10-24 13:02:11.050015	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
8	2019-10-24 13:02:12.066082	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Solicitation for fe80::2be:75ff:fef6:1d8e from 4c:4e:35:fc:fc:d8
9	2019-10-24 13:02:12.066234	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	82	Neighbor Advertisement fe80::2be:75ff:fef6:1d8e (rtr, sol)
10	2019-10-24 13:02:12.998422	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
11	2019-10-24 13:02:13.002105	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
12	2019-10-24 13:02:14.090251	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
13	2019-10-24 13:02:14.998544	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
14	2019-10-24 13:02:15.178350	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
15	2019-10-24 13:02:17.059963	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	90	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8 from 00:be:75:f6:1d:8e
16	2019-10-24 13:02:17.062512	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	82	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

Pontos principais:

1. O firewall envia uma mensagem de solicitação de vizinho IPv6 que solicita o endereço MAC do dispositivo upstream (IP fc00:1:1:2::2).
2. O roteador responde com um anúncio de vizinho IPv6.
3. O firewall envia uma solicitação de eco ICMP IPv6.
4. O dispositivo upstream (roteador fc00:1:1:2::2) envia uma mensagem de solicitação de vizinho IPv6 que solicita o endereço MAC do endereço IPv6 fc00:1:1:1::100.
5. O firewall envia uma solicitação de eco ICMP IPv6 adicional.
6. O roteador upstream envia uma mensagem adicional de solicitação de vizinhos IPv6 que solicita o endereço MAC do endereço IPv6 fc00:1:1:1::100.

O ponto 4 é muito interessante. Normalmente, o roteador upstream solicita o MAC da interface EXTERNA do firewall (fc00:1:1:2::2), mas, em vez disso, solicita o fc00:1:1:1::100. Essa é uma indicação de um erro de configuração.

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Verifique a tabela de vizinhos IPv6.

A tabela de vizinhos IPv6 do firewall está preenchida corretamente.

<#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
```

Ação 2. Verifique a configuração do IPv6.

Essa é a configuração do 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
```

A configuração do dispositivo upstream revela o erro de configuração:

<#root>

```
Router#
```

show run interface g0/0.202

```
!
interface GigabitEthernet0/0.202
 encapsulation dot1Q 202
 vrf forwarding VRF202
 ip address 192.168.2.72 255.255.255.0
 ipv6 address FC00:1:1:2::2
```

/48

Capturas - Cenário Funcional

A alteração da máscara de sub-rede (de /48 para /64) corrigiu o problema. Essa é a captura CAPI no cenário funcional.

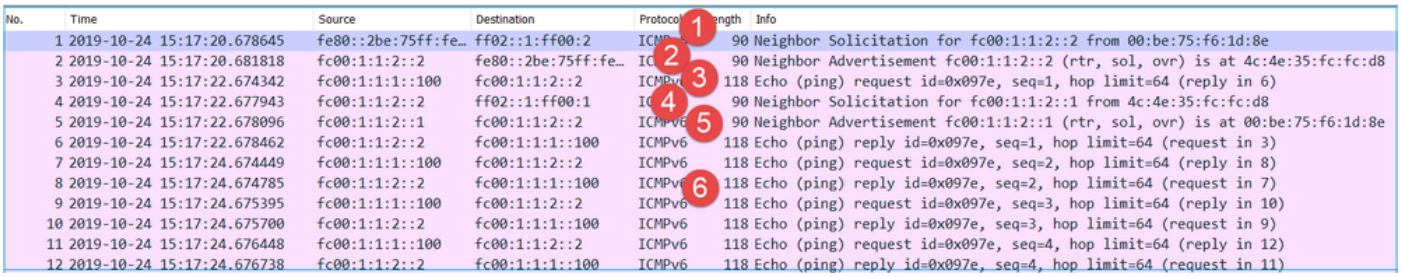
No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 15:17:20.677775	fc00:1:1:1::100	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
2	2019-10-24 15:17:20.677989	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6	86	Neighbor Advertisement fc00:1:1:1::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:ae
3	2019-10-24 15:17:20.678401	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=0, hop limit=64 (no response found!)
4	2019-10-24 15:17:22.674281	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=1, hop limit=64 (no response found!)
5	2019-10-24 15:17:24.674403	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=2, hop limit=64 (reply in 6)
6	2019-10-24 15:17:24.674815	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	114	Echo (ping) reply id=0x097e, seq=2, hop limit=64 (request in 5)
7	2019-10-24 15:17:24.675242	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=3, hop limit=64 (reply in 8)
8	2019-10-24 15:17:24.675731	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	114	Echo (ping) reply id=0x097e, seq=3, hop limit=64 (request in 7)
9	2019-10-24 15:17:24.676356	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	114	Echo (ping) request id=0x097e, seq=4, hop limit=64 (reply in 10)
10	2019-10-24 15:17:24.676753	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	114	Echo (ping) reply id=0x097e, seq=4, hop limit=64 (request in 9)

Ponto-chave:

1. O roteador envia uma mensagem de solicitação de vizinho IPv6 que solicita o endereço

MAC do dispositivo upstream (IP fc00:1:1:1::1).
2. O firewall responde com um anúncio de vizinho IPv6.
3. O roteador envia solicitações de eco ICMP e obtém respostas de eco.

Conteúdo do CAPO:



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

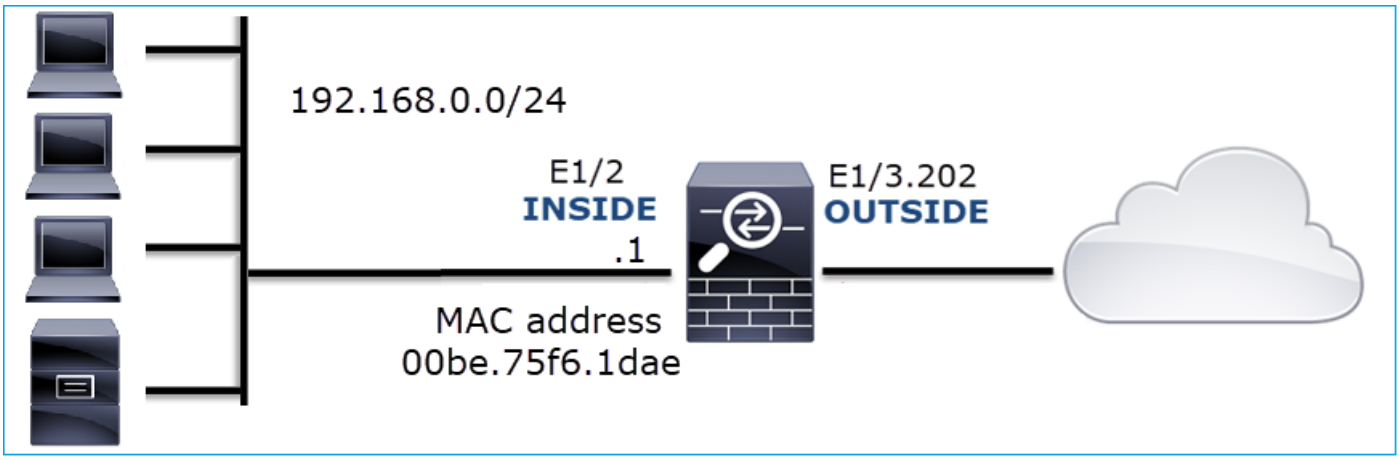
Pontos principais:

1. O firewall envia uma mensagem de solicitação de vizinho IPv6 que solicita o endereço MAC do dispositivo upstream (IP fc00:1:1:2::2).
2. O firewall responde com um anúncio de vizinho IPv6.
3. O firewall envia uma solicitação de eco ICMP.
4. O roteador envia uma mensagem de solicitação de vizinho IPv6 que solicita o endereço MAC do dispositivo downstream (IP fc00:1:1:1::1).
5. O firewall responde com um anúncio de vizinho IPv6.
6. O firewall envia solicitações de eco ICMP e obtém respostas de eco.

Caso 12. Problema de conectividade intermitente (envenenamento ARP)

Descrição do problema: os hosts internos (192.168.0.x/24) têm problemas de conectividade intermitentes com os hosts na mesma sub-rede

Esta imagem mostra a topologia:



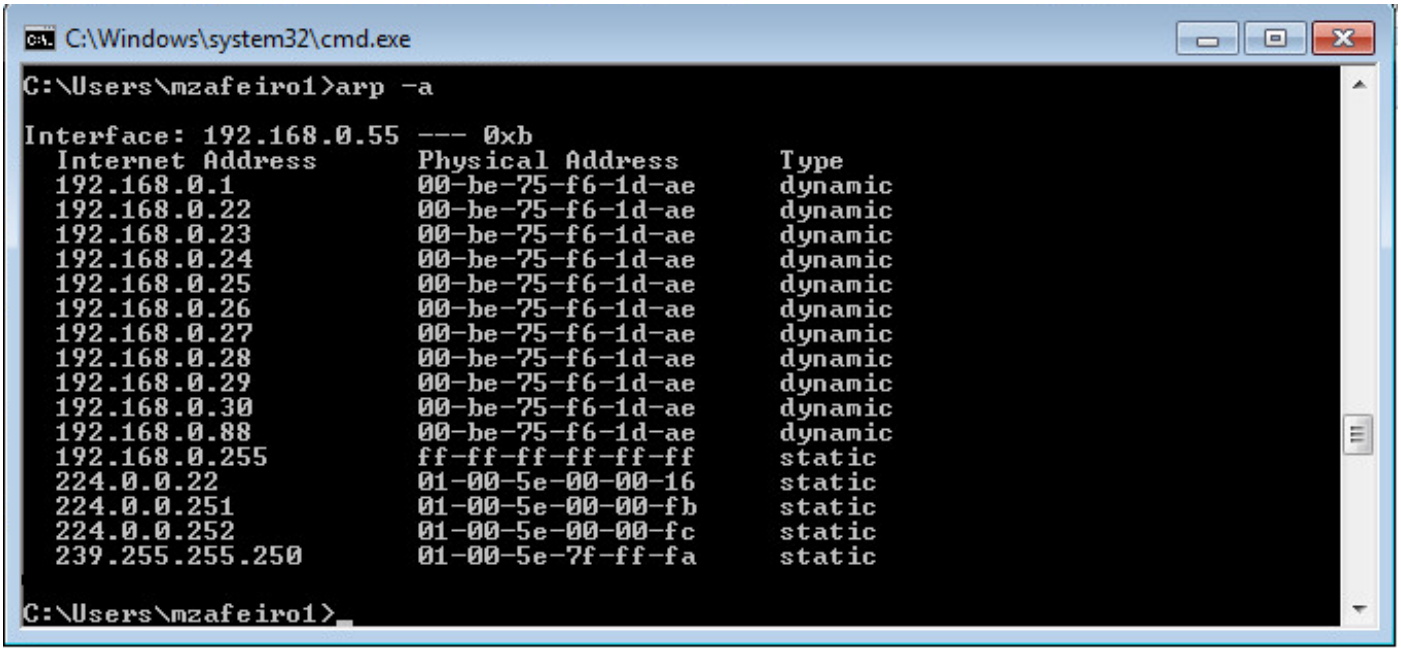
Fluxo afetado:

IP orig.: 192.168.0.x/24

IP do Horário de Verão: 192.168.0.x/24

Protocolo: qualquer

O cache ARP de um host interno parece estar inviabilizado:



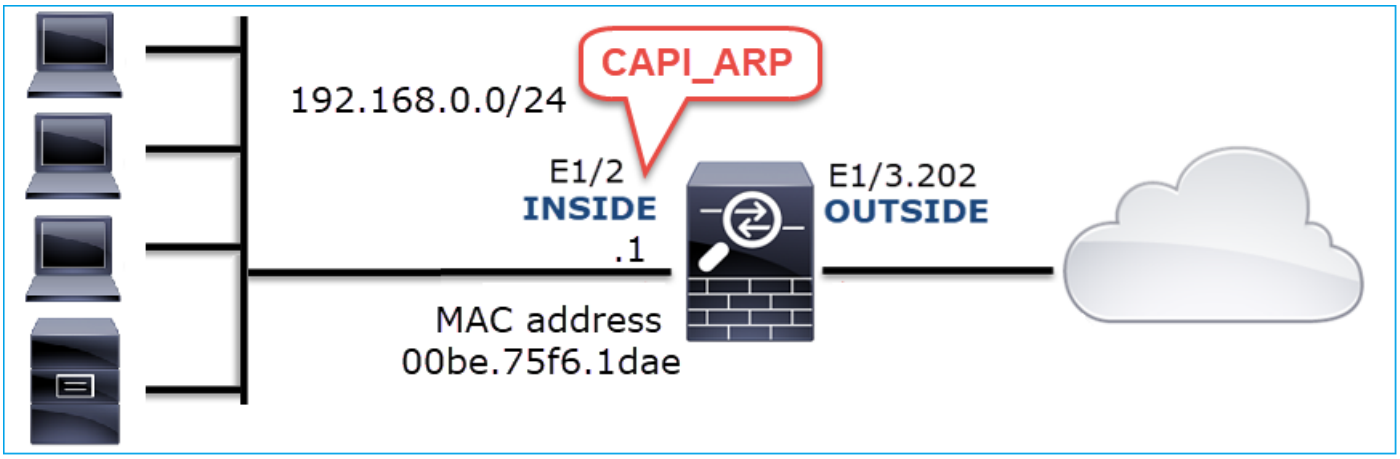
Capturar análise

Habilitar uma captura no mecanismo LINA do FTD

Essa captura só captura pacotes ARP na interface INSIDE:

<#root>

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



Capturas - cenário não funcional:

A captura na interface INSIDE do firewall contém.

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

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

Pontos principais:

1. O firewall recebe várias solicitações ARP para IPs dentro da rede 192.168.0.x/24
2. O firewall responde a todos eles (proxy-ARP) com seu próprio endereço MAC

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Verifique a configuração do NAT.

Com relação à configuração do NAT, há casos em que a palavra-chave no-proxy-arp pode impedir o comportamento anterior:

<#root>

```
firepower#
```

show run nat

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

```
no-proxy-arp
```

Ação 2. Desative a funcionalidade proxy-arp na interface do firewall.

Se a palavra-chave ‘no-proxy-arp’ não resolver o problema, tente desativar o proxy ARP na própria interface. No caso de FTD, no momento da elaboração deste documento, você precisa usar o FlexConfig e implantar o comando (especifique o nome da interface apropriada).

```
sysopt noproxyarp INSIDE
```

Caso 13. Identificar Identificadores de Objeto (OIDs - Object Identifiers) SNMP que causam problemas na CPU

Esse caso demonstra como determinados OIDs SNMP para polling de memória foram identificados como a causa raiz de hogs de CPU (problema de desempenho) com base na análise de capturas de pacotes SNMP versão 3 (SNMPv3).

Descrição do problema: as sobrecargas nas interfaces de dados aumentam continuamente. Pesquisas adicionais revelaram que também há monopolizadores de CPU (causados pelo processo SNMP) que são a causa raiz das sobrecargas da interface.

A próxima etapa no processo de identificação e solução de problemas foi identificar a causa raiz dos hogs de CPU causados pelo processo SNMP e, em particular, restringir o escopo do problema para identificar os Identificadores de Objetos (OID) SNMP que, quando interrogados, poderiam potencialmente resultar em hogs de CPU.

Atualmente, o mecanismo LINA do FTD não fornece um comando 'show' para OIDs SNMP que são pesquisados em tempo real.

A lista de OIDs de SNMP para polling pode ser recuperada da ferramenta de monitoramento de SNMP, no entanto, neste caso, houve estes fatores preventivos:

- O administrador do FTD não teve acesso à ferramenta de monitoramento SNMP
- O SNMP versão 3 com autenticação e criptografia de dados para privacidade foi configurado no FTD

Capturar análise

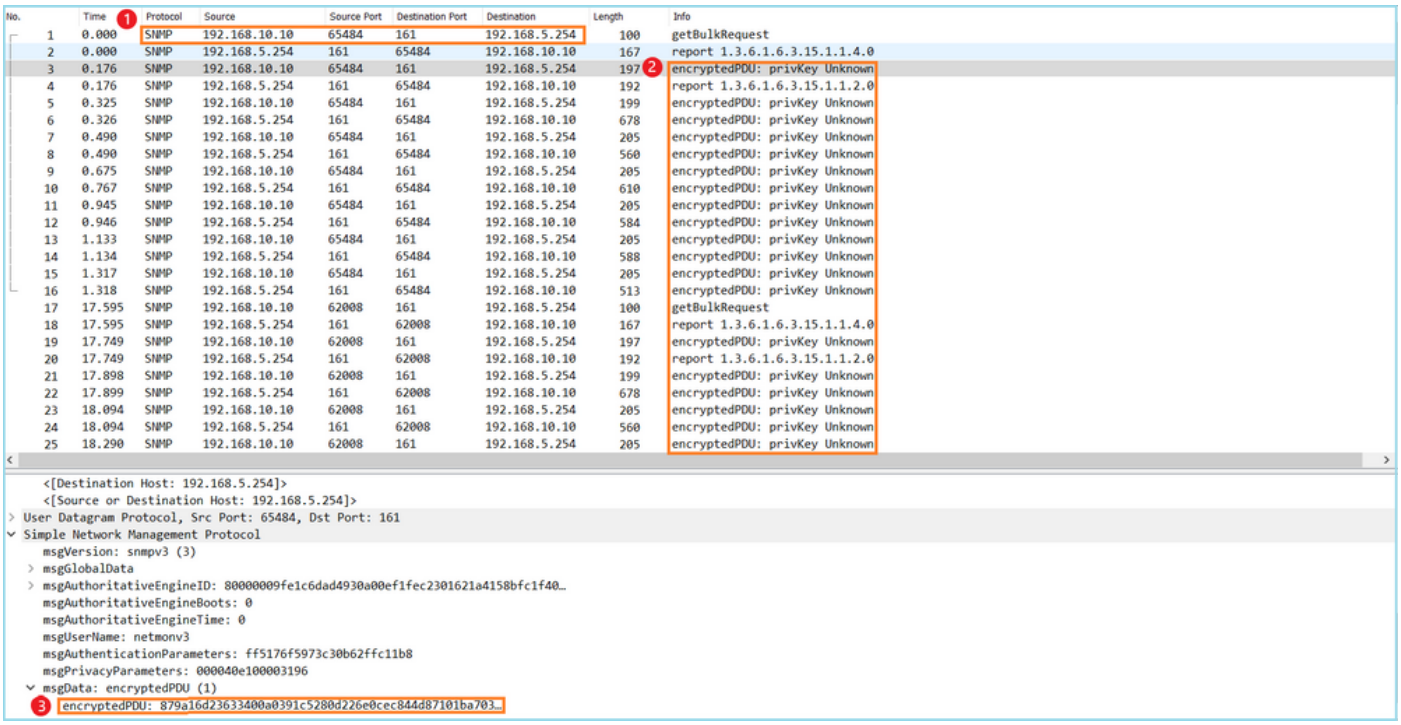
Como o administrador do FTD tinha as credenciais para a autenticação e a criptografia de dados do SNMP versão 3, este plano de ação foi proposto:

1. Tirar capturas de pacotes SNMP
2. Salve as capturas e use as preferências do protocolo SNMP Wireshark para especificar as

credenciais da versão 3 do SNMP para descriptografar os pacotes da versão 3 do SNMP. As capturas descriptografadas são usadas para a análise e recuperação de OIDs SNMP

Configure as capturas de pacotes SNMP na interface usada na configuração do host do servidor SNMP:

```
<#root>

firepower#

show run snmp-server  | include host

snmp-server host management 192.168.10.10 version 3 netmonv3


firepower#

show ip address management

System IP Address:
Interface                Name                   IP address      Subnet mask     Method
Management0/0            management             192.168.5.254   255.255.255.0   CONFIG
Current IP Address:
Interface                Name                   IP address      Subnet mask     Method
Management0/0            management             192.168.5.254   255.255.255.0   CONFIG

firepower#

capture capsnmp interface management buffer 10000000 match udp host 192.168.10.10  host 192.168.5.254 eq


firepower#

show capture capsnmp


capture capsnmp type raw-data buffer 10000000 interface outside [Capturing -

9512

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



Pontos principais:

1. Endereços/portas SNMP de origem e destino.
2. Não foi possível decodificar a PDU do protocolo SNMP porque privKey é desconhecido para o Wireshark.
3. O valor da primitiva encryptedPDU.

Ações recomendadas

As ações listadas nesta seção têm como objetivo restringir ainda mais o problema.

Ação 1. Descriptografe as capturas SNMP.

Salve as capturas e edite as preferências do protocolo SNMP Wireshark para especificar as credenciais da versão 3 do SNMP para descriptografar os pacotes.

<#root>

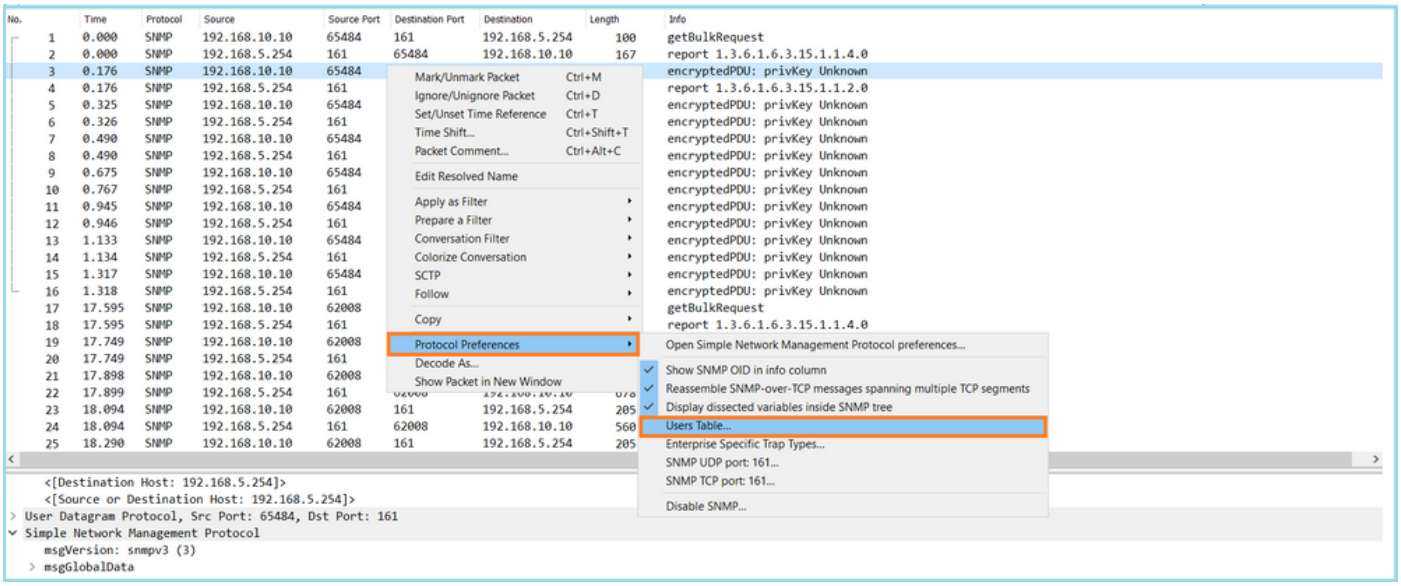
```
firepower#
```

copy /pcap capture: tftp:

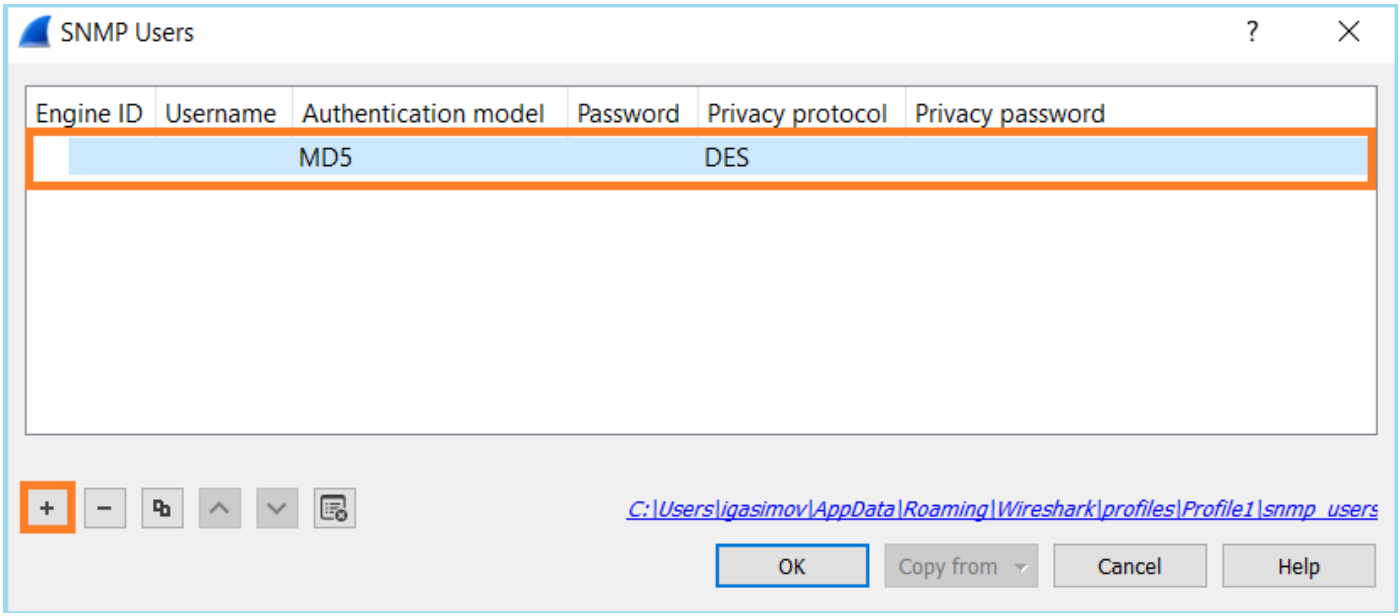
```
Source capture name [capsnmp]?

Address or name of remote host []? 192.168.10.253

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

!!!!!!
64 packets copied in 0.40 secs

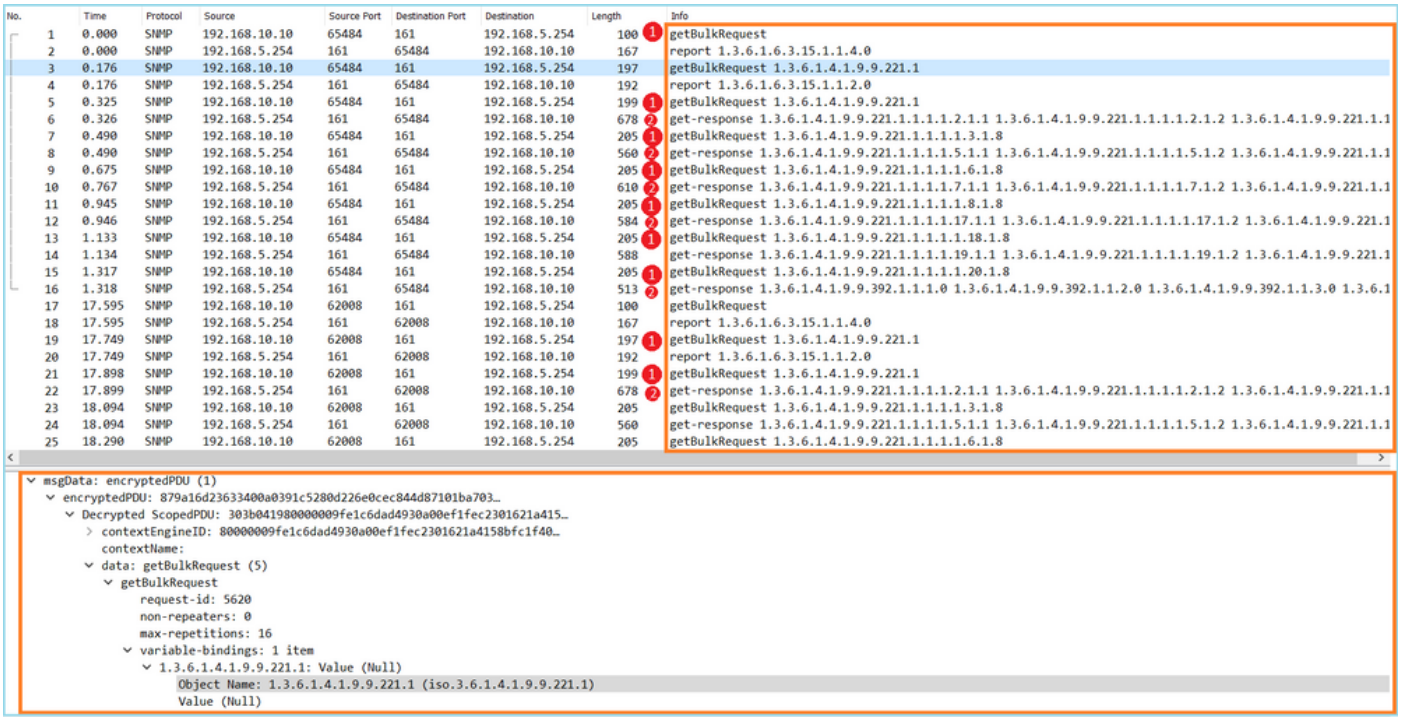
Abra o arquivo de captura no Wireshark, selecione um pacote SNMP e navegue para Protocol Preferences > Users Table, como mostrado na imagem:



Na tabela Usuários SNMP, foram especificados o nome de usuário, o modelo de autenticação, a senha de autenticação, o protocolo de privacidade e a senha de privacidade do SNMP versão 3 (as credenciais reais não são mostradas abaixo):



Quando as configurações dos usuários do SNMP foram aplicadas, o Wireshark mostrou PDUs SNMP descriptografadas:



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

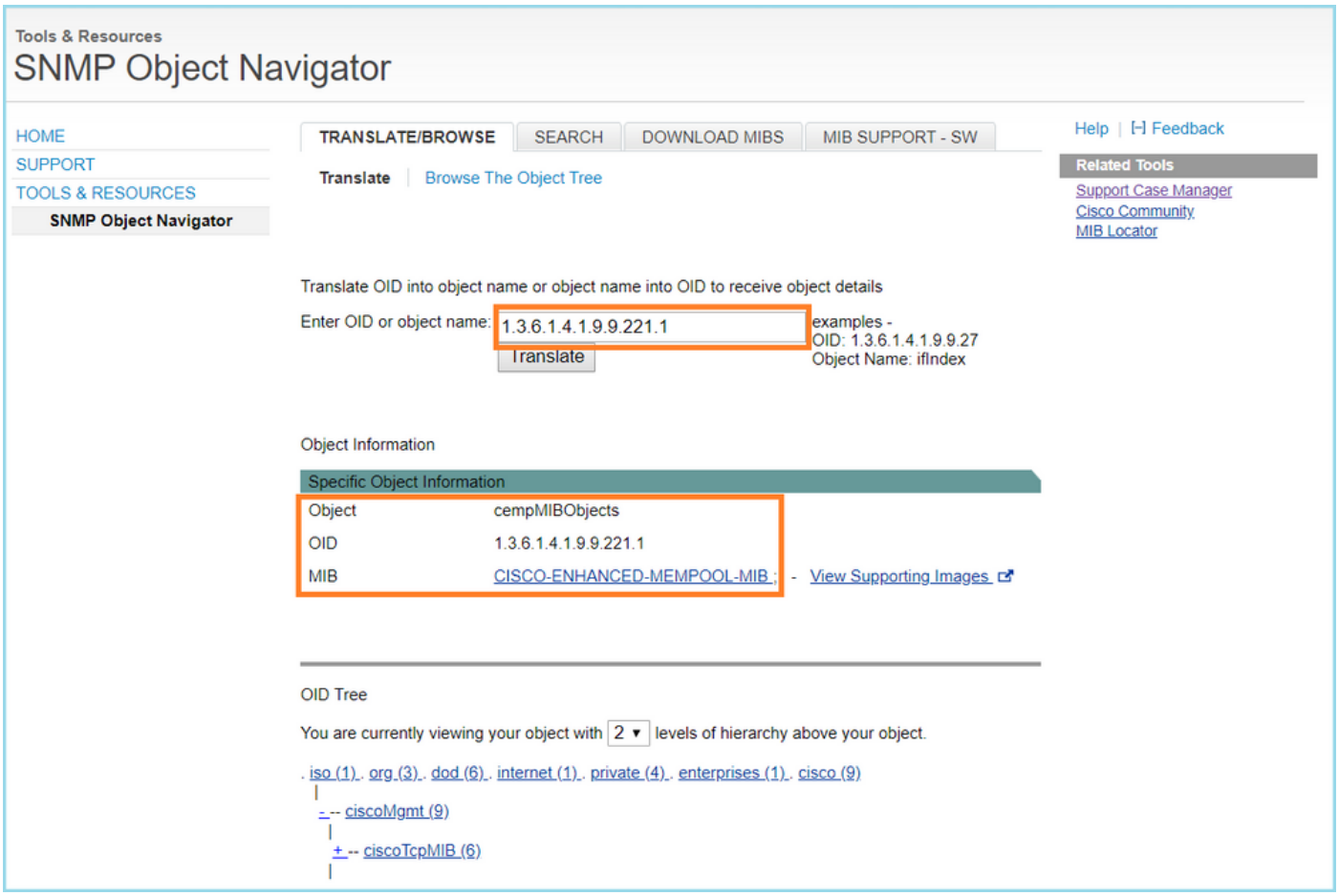
```
msgData: encryptedPDU (1)
  encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703…
  Decrypted ScopedPDU: 303b041980000009fe1c6dad4930a00ef1fec2301621a415…
    contextEngineID: 80000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40…
    contextName:
    data: getBulkRequest (5)
      getBulkRequest
        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)
```

Pontos principais:

1. As ferramentas de monitoramento do SNMP usaram o SNMP getBulkRequest para consultar e percorrer o OID pai 1.3.6.1.4.1.9.9.221.1 e os OIDs relacionados.
2. O FTD respondeu a cada getBulkRequest com get-response que contém OIDs relacionados a 1.3.6.1.4.1.9.9.221.1.

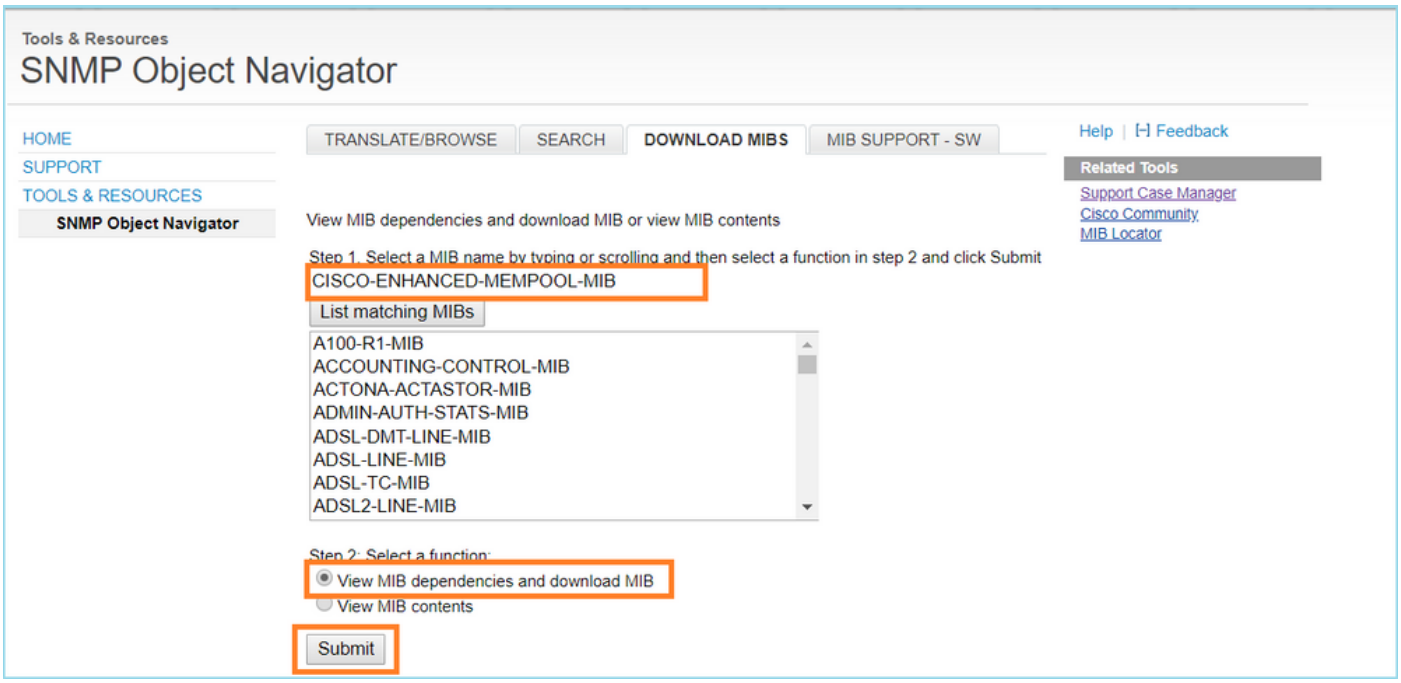
Ação 2. Identificar os OIDs SNMP.

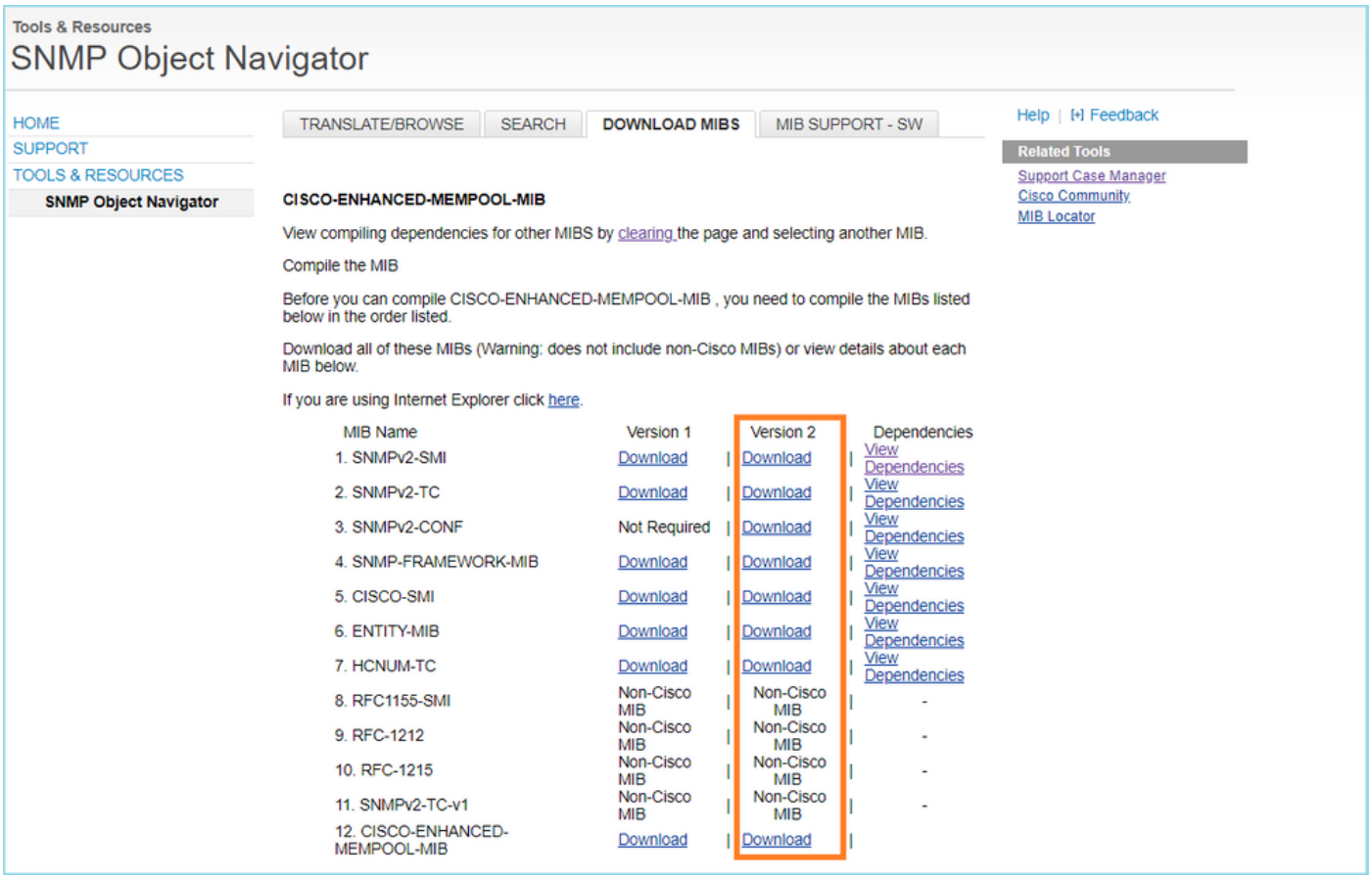
O SNMP Object Navigator mostrou que o OID 1.3.6.1.4.1.9.9.221.1 pertence à base de informações de gerenciamento (MIB) chamada CISCO-ENHANCED-MEMPOOL-MIB, como mostrado na imagem:



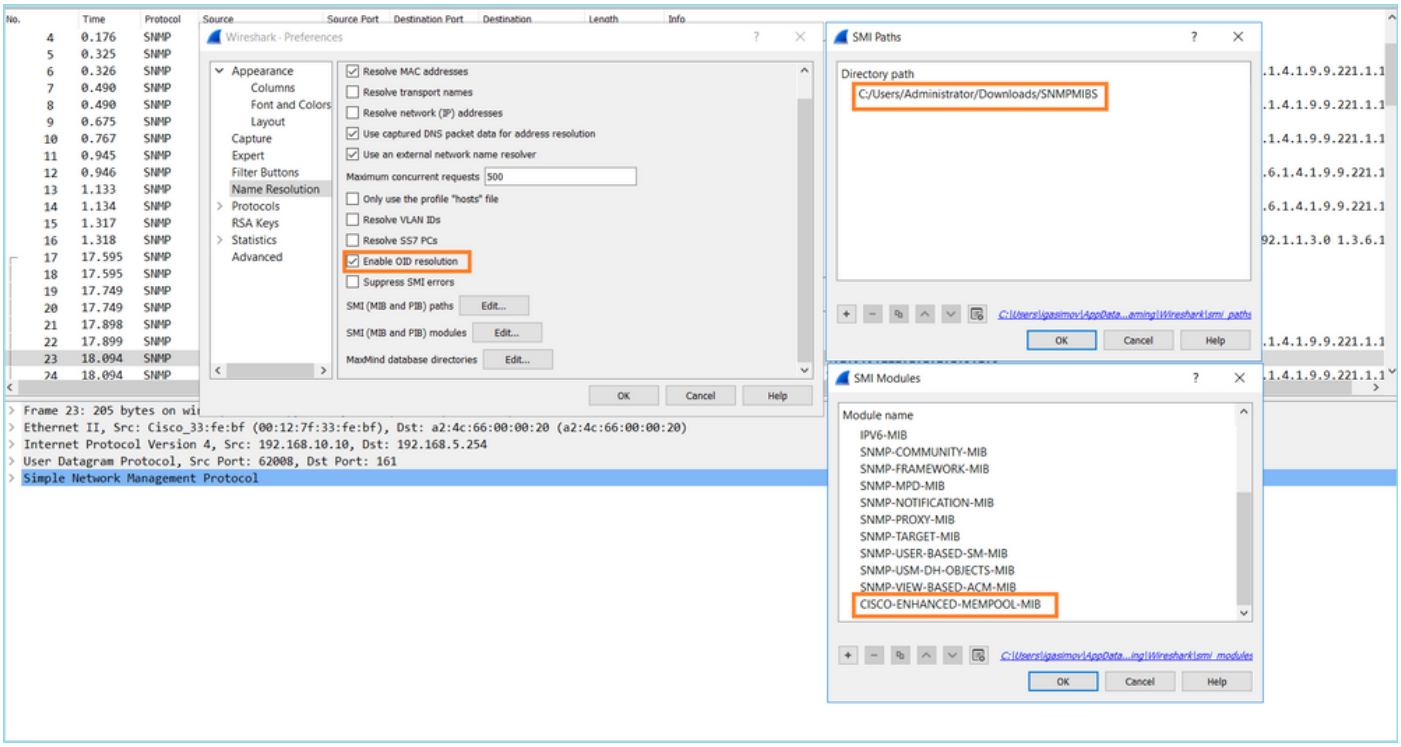
Para exibir os OIDs em formato legível no Wireshark:

1. Baixe o MIB CISCO-ENHANCED-MEMPOOL-MIB e suas dependências, como mostrado na imagem:

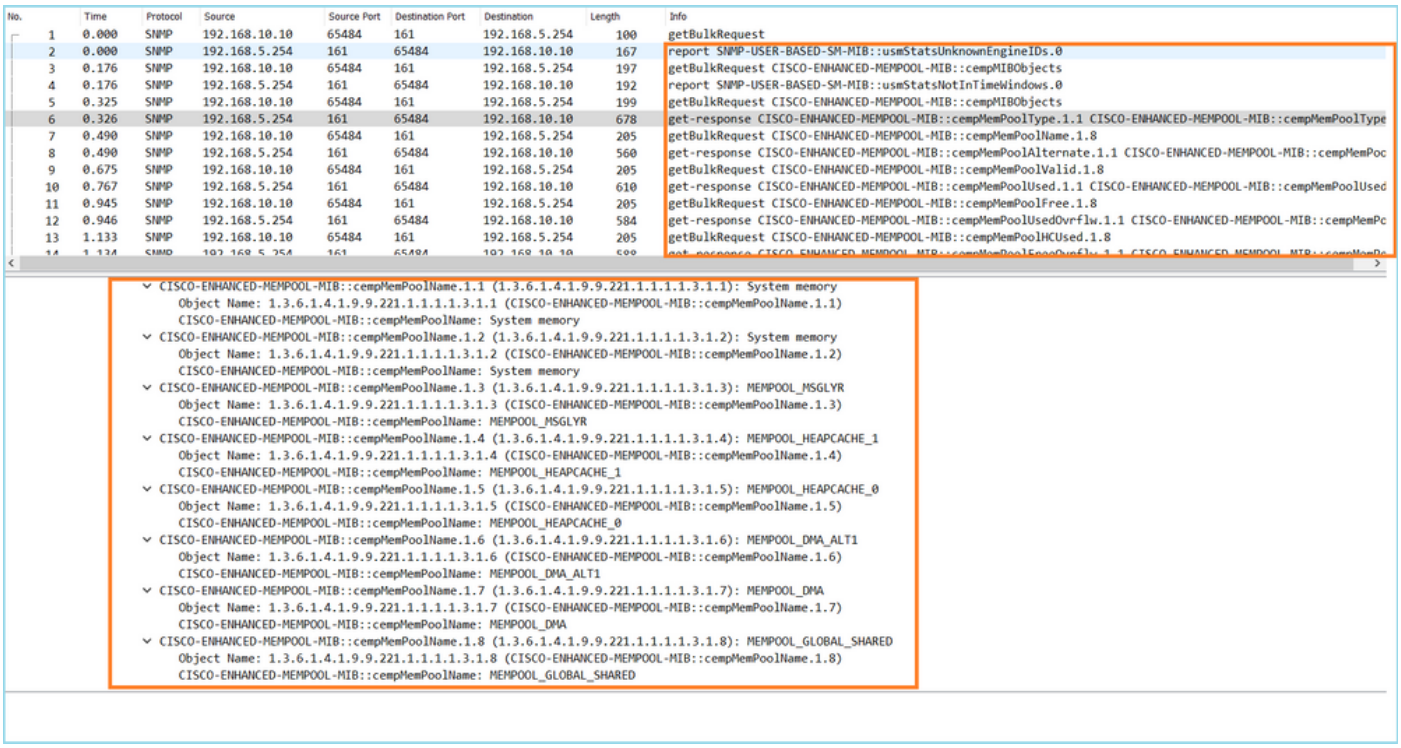




2. No Wireshark, na janela Edit > Preferences > Name Resolution, a opção Enable OID Resolution está marcada. Na janela SMI (MIB e caminhos PIB), especifique a pasta com os MIBs baixados e em SMI (MIB e módulos PIB). O CISCO-ENHANCED-MEMPOOL-MIB é adicionado automaticamente à lista de módulos:



3. Quando o Wireshark for reiniciado, a resolução do OID será ativada:



Com base na saída descriptografada do arquivo de captura, a ferramenta de monitoramento SNMP pesquisava periodicamente (intervalo de 10 segundos) dados sobre a utilização de pools de memória no FTD. Conforme explicado no artigo da Nota Técnica ASA SNMP Polling for Memory-Related Statistics , a pesquisa da utilização do Global Shared Pool (GSP) com SNMP resulta em alto uso da CPU. Nesse caso, a partir das capturas, ficou claro que a utilização do Pool compartilhado global foi sondada periodicamente como parte do SNMP getBulkRequest primitivo.

Para minimizar os hogs de CPU causados pelo processo SNMP, foi recomendado seguir as etapas de mitigação para os Hogs de CPU para SNMP mencionados no artigo e evitar pesquisar os OIDs relacionados ao GSP. Sem a pesquisa de SNMP para os OIDs relacionados ao GSP, não foram observados hogs de CPU causados pelo processo SNMP e a taxa de saturação diminuiu significativamente.

Informações Relacionadas

- Guias de configuração do Cisco Firepower Management Center
- Esclarecer as ações da regra de política de controle de acesso do Firepower Threat Defense
- Trabalhe com capturas do Firepower Threat Defense e do Packet Tracer
- Aprenda o Wireshark

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.