

Analysieren von Firepower-Firewall-Erfassungen zur effektiven Behebung von Netzwerkproblemen

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Sammeln und Exportieren von Aufzeichnungen aus der NGFW-Produktfamilie](#)

[Erfassen von FXOS-Inhalten](#)

[FTD-Lina-Erfassungen aktivieren und erfassen](#)

[Aktivieren und Sammeln von FTD Snort-Erfassungen](#)

[Fehlerbehebung](#)

[Fall 1: Kein TCP-SYN an der Ausgangsschnittstelle](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Mögliche Ursachen und empfohlene Aktionen - Zusammenfassung](#)

[Fall 2: TCP SYN vom Client, TCP RST vom Server](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 3: TCP 3-Wege-Handshake + RST von einem Endgerät](#)

[Erfassungsanalyse](#)

[3.1 - TCP-3-Wege-Handshake + verzögerte RST vom Client](#)

[Empfohlene Maßnahmen](#)

[3.2 - TCP-3-Wege-Handshake + verzögerte FIN/ACK vom Client + verzögerte RST vom Server](#)

[Empfohlene Maßnahmen](#)

[3.3 - TCP-3-Wege-Handshake + verzögerte RST vom Client](#)

[Empfohlene Maßnahmen](#)

[3.4 - TCP-3-Wege-Handshake + sofortige RST vom Server](#)

[Empfohlene Maßnahmen](#)

[Fall 4: TCP RST vom Client](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 5: Langsame TCP-Übertragung \(Szenario 1\)](#)

[Szenario 1. Langsame Übertragung](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Szenario 2. Schnelle Übertragung](#)

[Fall 6: Langsame TCP-Übertragung \(Szenario 2\)](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 7: TCP-Verbindungsproblem \(Paketbeschädigung\)](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 8: UDP-Verbindungsproblem \(fehlende Pakete\)](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 9: HTTPS-Verbindungsproblem \(Szenario 1\)](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 10: HTTPS-Verbindungsproblem \(Szenario 2\)](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 11: IPv6-Verbindungsproblem](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 12: Intermittierendes Verbindungsproblem \(ARP Poisoning\)](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Fall 13: Identifizieren von SNMP-Objektbezeichnern \(OIDs\), die CPU-Hogs verursachen](#)

[Erfassungsanalyse](#)

[Empfohlene Maßnahmen](#)

[Zugehörige Informationen](#)

Einleitung

In diesem Dokument werden verschiedene Verfahren zur Analyse von Paketerfassungen beschrieben, die der effektiven Fehlersuche bei Netzwerkproblemen dienen.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- FirePOWER Plattformarchitektur
- NGFW-Protokolle
- NGFW Packet-Tracer

Bevor Sie mit der Analyse der Paketerfassung beginnen, sollten Sie außerdem folgende Anforderungen erfüllen:

- **Protokollvorgang kennen** - Beginnen Sie nicht mit der Überprüfung einer Paketerfassung, wenn Sie nicht wissen, wie das erfasste Protokoll funktioniert.
- **Kenne die Topologie** - Du musst die End-to-End-Transitgeräte kennen. Wenn dies nicht möglich ist, müssen Sie zumindest die vor- und nachgeschalteten Geräte kennen.
- **Kennen Sie die Appliance** - Sie müssen wissen, wie Ihr Gerät mit Paketen umgeht, welche Schnittstellen beteiligt sind (Eingang/Ausgang), welche Gerätearchitektur verwendet wird und welche Erfassungspunkte vorhanden sind.
- **Konfiguration kennen** - Sie müssen wissen, wie ein Paketfluss vom Gerät behandelt werden soll, und zwar in Bezug auf:
 - Routing-/Ausgangsschnittstelle
 - Anwendung von Richtlinien
 - Network Address Translation (NAT)
- **Kenntnis der verfügbaren Tools** - Neben den Erfassungen wird empfohlen, andere Tools und Techniken (wie Protokollierung und Tracer) anzuwenden und diese bei Bedarf mit den erfassten Paketen zu korrelieren.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Die meisten Szenarien basieren auf FP4140 mit FTD-Software 6.5.x.
- FMC mit Software 6.5.x

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Die Paketerfassung zählt zu den am häufigsten übersehenen Tools zur Fehlerbehebung, die derzeit verfügbar sind. Das Cisco TAC löst täglich viele Probleme durch die Analyse der erfassten Daten.

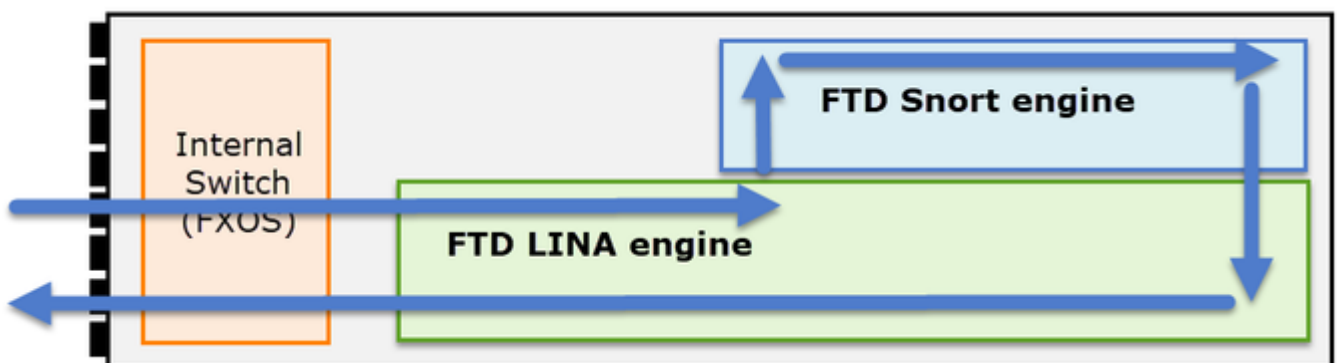
Das vorliegende Dokument soll Netzwerk- und Sicherheitstechniker dabei unterstützen, gängige Netzwerkprobleme zu identifizieren und zu beheben, die hauptsächlich auf einer Paketerfassungsanalyse basieren.

Alle in diesem Dokument vorgestellten Szenarien basieren auf tatsächlichen Benutzerfällen aus dem Cisco Technical Assistance Center (TAC).

Das Dokument behandelt die Paketerfassung aus der Sicht der Cisco Next-Generation Firewall (NGFW). Die gleichen Konzepte gelten jedoch auch für andere Gerätetypen.

Sammeln und Exportieren von Aufzeichnungen aus der NGFW-Produktfamilie

Bei einer FirePOWER-Appliance (1xxx, 21xx, 41xx, 93xx) und einer FirePOWER Threat Defense (FTD)-Anwendung kann eine Paketverarbeitung wie im Bild dargestellt dargestellt dargestellt dargestellt werden.



1. Ein Paket gelangt an die Eingangsschnittstelle, und es wird vom internen Chassis-Switch verarbeitet.
2. Das Paket gelangt in die FTD Lina-Engine, die hauptsächlich L3/L4-Prüfungen durchführt.
3. Wenn die Richtlinie erfordert, wird das Paket von der Snort-Engine geprüft (hauptsächlich L7-Inspektion).
4. Die Snort-Engine gibt ein Urteil für das Paket zurück.
5. Die LINA-Engine verwirft oder leitet das Paket basierend auf dem Urteil von Snort weiter.
6. Das Paket gelangt über den internen Chassis-Switch aus dem Chassis.

Basierend auf der gezeigten Architektur können die FTD-Aufnahmen an drei (3) verschiedenen Orten gemacht werden:

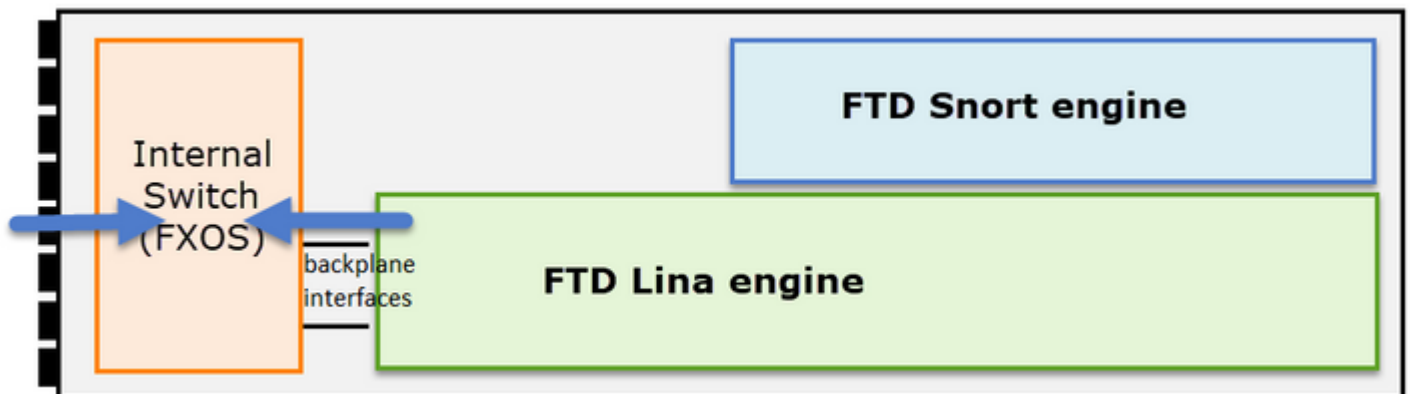
- FXOS
- FTD Lina-Engine
- FTD Snort-Engine

Erfassen von FXOS-Inhalten

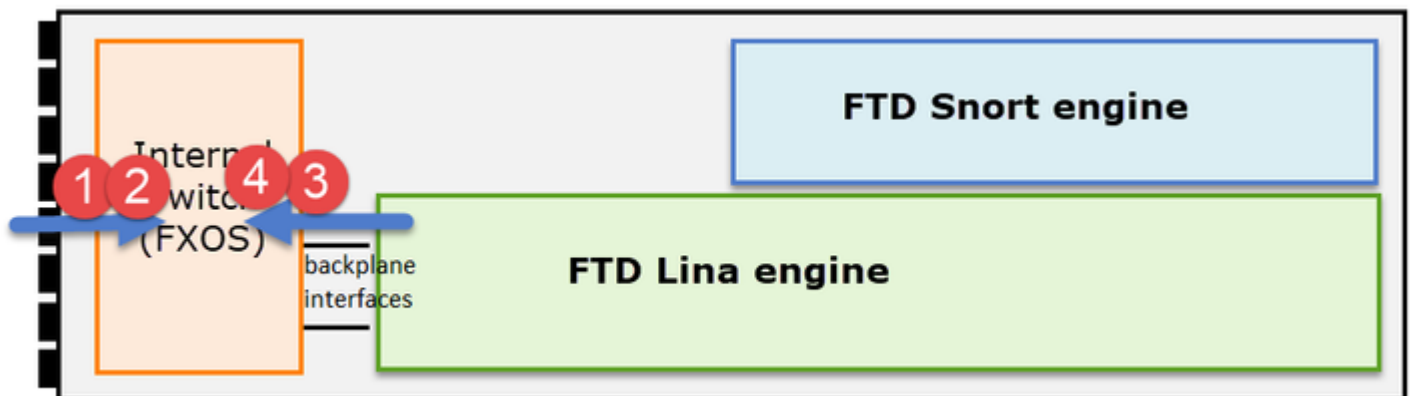
Der Prozess wird in diesem Dokument beschrieben:

https://www.cisco.com/c/en/us/td/docs/security/firepower/fxos/fxos271/web-guide/b_GUI_FXOS_ConfigGuide_271/troubleshooting.html#concept_E8823CC63C934A909BBC0DF12F301DE

FXOS-Aufnahmen können aus Sicht des internen Switches nur in Eingangsrichtung gemacht werden.



Hier sind zwei Erfassungspunkte pro Richtung dargestellt (aufgrund der internen Switch-Architektur).



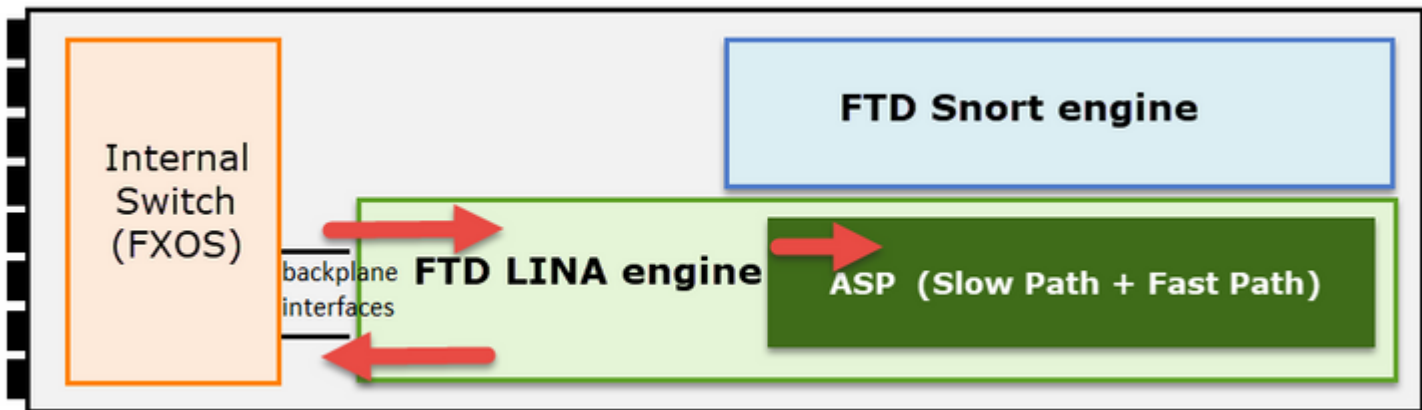
Die in den Punkten 2, 3 und 4 erfassten Pakete weisen ein Virtual Network Tag (VNTag) auf.

Hinweis: FXOS-Aufnahmen auf Chassis-Ebene sind nur auf den Plattformen FP41xx und FP93xx verfügbar. FP1xxx und FP21xx bieten diese Funktion nicht.

FTD-Lina-Erfassungen aktivieren und erfassen

Wichtigste Punkte:

- Eingangsschnittstelle
- Ausgangs-Schnittstelle
- Accelerated Security Path (ASP)



Sie können entweder die Firepower Management Center-Benutzeroberfläche (FMC UI) oder FTD CLI verwenden, um die FTD-Lina-Aufnahmen zu aktivieren und zu sammeln.

Aktivieren Sie die Erfassung über die CLI auf der INSIDE-Schnittstelle:

```
<#root>
firepower#
capture CAPI interface INSIDE match icmp host 192.168.103.1 host 192.168.101.1
```

Diese Erfassung gleicht den Datenverkehr zwischen den IP-Adressen 192.168.103.1 und 192.168.101.1 in beide Richtungen ab.

Aktivieren Sie die ASP-Erfassung, um alle Pakete anzuzeigen, die vom FTD Lina-Modul verworfen wurden:

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

FTD-Lina-Aufzeichnung auf einen FTP-Server exportieren:

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

FTD-Lina-Aufzeichnung auf einen TFTP-Server exportieren:

```
<#root>
firepower#
```

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

Ab der Version FMC 6.2.x können Sie FTD Lina Captures von der FMC UI aus aktivieren und sammeln. Eine andere Möglichkeit, FTD-Aufnahmen von einer FMC-verwalteten Firewall zu sammeln, ist diese.

Schritt 1

Bei LINA- oder ASP-Erfassung kopieren Sie die Erfassung auf die FTD-Diskette.

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

```
Source capture name [capin]?
Destination filename [capin.pcap]?
!!!!
```

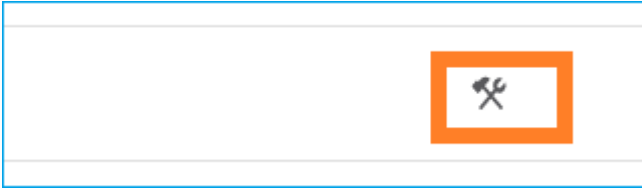
Schritt 2

Navigieren Sie in den Expertenmodus, suchen Sie die gespeicherte Aufzeichnung, und kopieren Sie sie in den Speicherort /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
```

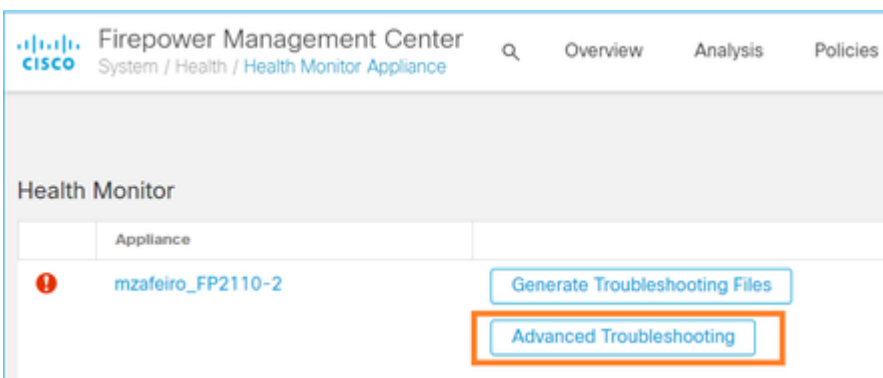
Schritt 3

Melden Sie sich beim FMC an, das die FTD verwaltet, und navigieren Sie zu **Devices (Geräte) > Device Management (Geräteverwaltung)**. Suchen Sie das FTD-Gerät, und wählen Sie das Symbol **Fehlerbehebung**:

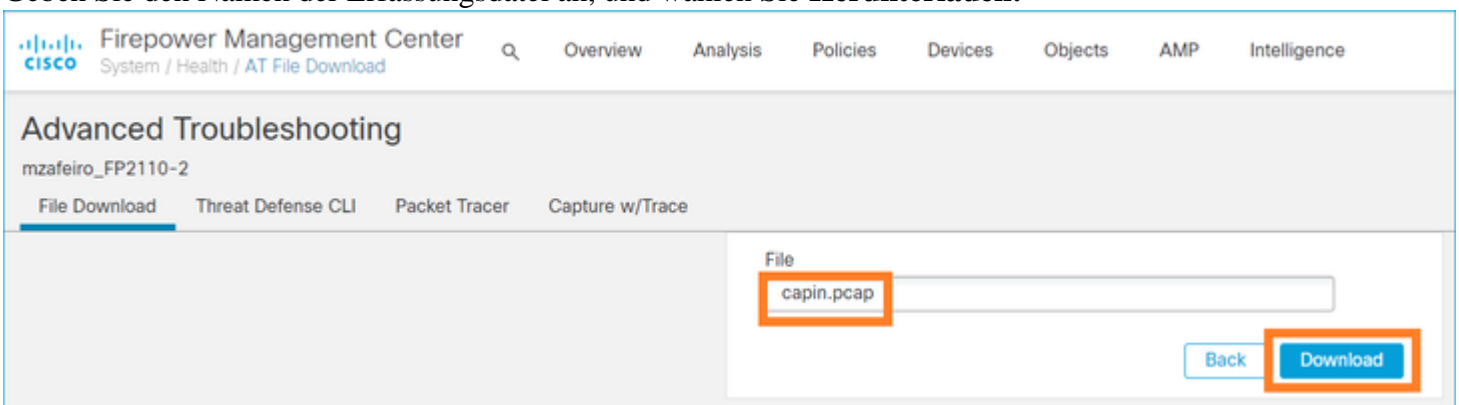


Schritt 4

Erweiterte Fehlerbehebung auswählen:



Geben Sie den Namen der Erfassungsdatei an, und wählen Sie **Herunterladen**:

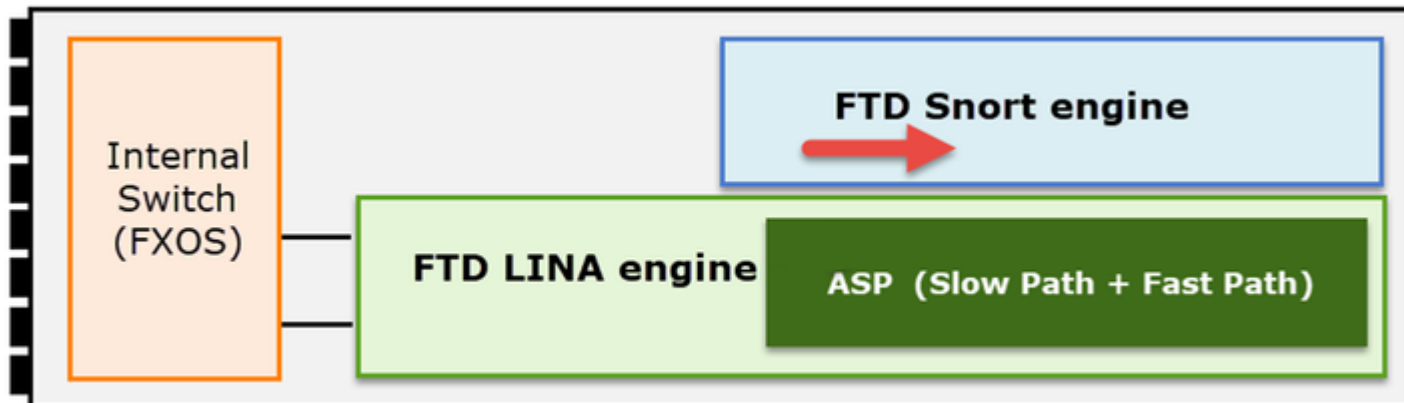


Weitere Beispiele zum Aktivieren/Sammeln von Erfassungen über die FMC-Benutzeroberfläche finden Sie in diesem Dokument:

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

Aktivieren und Sammeln von FTD Snort-Erfassungen

Der Fangpunkt ist hier im Bild dargestellt.



Snapshot-Erfassung aktivieren:

```
<#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
```

So schreiben Sie die Aufzeichnung in eine Datei mit dem Namen "capture.pcap" und kopieren sie per FTP auf einen Remote-Server:

```
<#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.
```

```
>
```

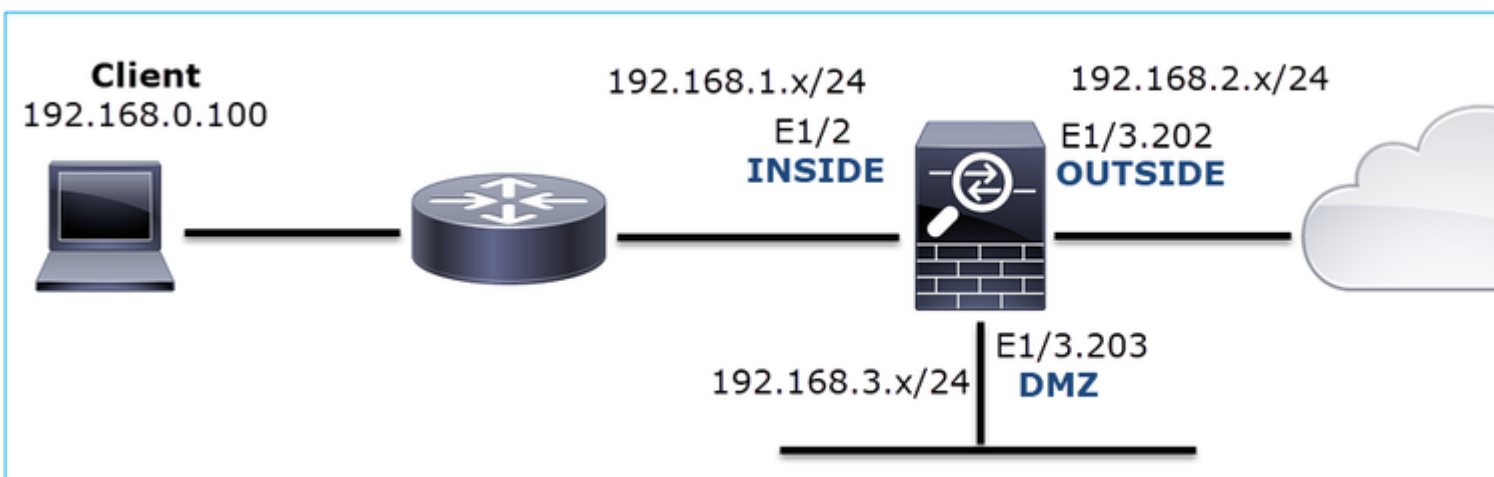
Weitere Erfassungsbeispiele auf Snort-Ebene mit verschiedenen Erfassungsfiltren finden Sie in diesem Dokument:

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

Fehlerbehebung

Fall 1: Kein TCP-SYN an der Ausgangsschnittstelle

Die Topologie ist im folgenden Bild dargestellt:



Problembeschreibung: HTTP funktioniert nicht

Betroffener Datenfluss:

Quelle IP: 192.168.0.100

Ziel-IP: 10.10.1.100

Protokoll: TCP 80

Erfassungsanalyse

Aktivieren Sie Aufnahmen auf der FTD LINA-Engine:

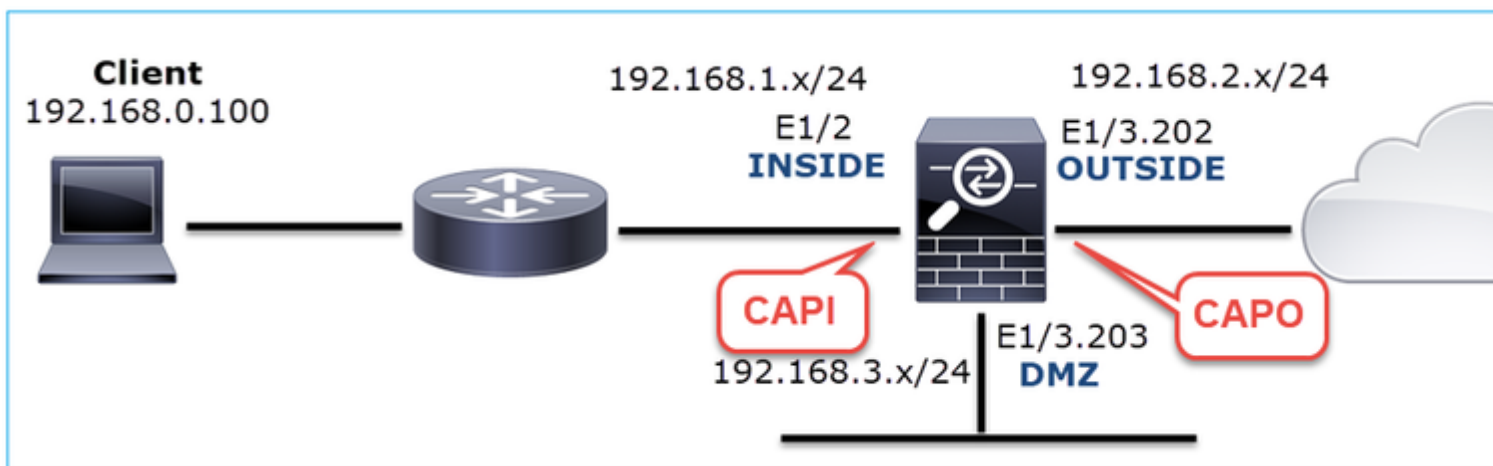
```
<#root>
```

```
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
```



Erfassungen - Funktionsszenario:

Als Basis ist es immer sehr nützlich, Aufzeichnungen aus einem funktionalen Szenario zu haben.

Erfassung auf NGFW INSIDE-Schnittstelle, wie in der Abbildung dargestellt:

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

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

Wichtigste Punkte:

1. TCP-3-Wege-Handshake
2. Bidirektionaler Datenaustausch.
3. Keine Verzögerungen zwischen den Paketen (basierend auf der Zeitdifferenz zwischen den Paketen)
4. Quell-MAC ist das richtige Downstream-Gerät.

Die Erfassung, die über die NGFW OUTSIDE-Schnittstelle erfolgt, wird im folgenden Bild dargestellt:

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

> Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
 > Ethernet II, Src: Cisco_f6:1d:8e (00:be:75:f6:1d:8e), Dst: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8)
 > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
 > Transmission Control Protocol, Src Port: 1779, Dst Port: 80, Seq: 0, Len: 0

Wichtigste Punkte:

1. Die gleichen Daten wie bei der CAPI-Erfassung.
2. Die Ziel-MAC ist das richtige Upstream-Gerät.

Erfassungen - Szenario ohne Funktion

Aus der Geräte-CLI sehen die Aufnahmen wie folgt aus:

```
<#root>
firepower#
show capture
capture CAPI type raw-data interface INSIDE
[Capturing - 484 bytes]
  match ip host 192.168.0.100 host 10.10.1.100
capture CAPO type raw-data interface OUTSIDE
[Capturing - 0 bytes]
  match ip host 192.168.0.100 host 10.10.1.100
```

CAPI-Inhalt:

```
<#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

```

Dies ist das Bild der CAPI-Erfassung in Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.100	10.10.1.100	TCP	66	3171 → 80 [SYN] Seq=0 Win=8192 Len=0
2	0.250420	192.168.0.100	10.10.1.100	TCP	66	3172 → 80 [SYN] Seq=0 Win=8192 Len=0
3	2.745781	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0
4	0.255074	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 3172 → 80 [SYN] Seq=0 Win=8192 Len=0
5	5.751883	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0
6	0.250070	192.168.0.100	10.10.1.100	TCP	62	[TCP Retransmission] 3172 → 80 [SYN] Seq=0 Win=8192 Len=0

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

Wichtigste Punkte:

1. Es werden nur TCP-SYN-Pakete angezeigt (kein TCP-3-Wege-Handshake).
2. Es gibt 2 TCP-Sitzungen (Quellport 3171 und 3172), die nicht eingerichtet werden können. Der Quell-Client sendet die TCP-SYN-Pakete erneut. Diese neu übertragenen Pakete werden von Wireshark als TCP-Neuübertragungen identifiziert.
3. Die TCP-Neuübertragungen erfolgen alle ~3, dann 6 usw. Sekunden.
4. Die Quell-MAC-Adresse stammt vom korrekten Downstream-Gerät.

Auf der Grundlage der beiden Aufzeichnungen kann der Schluss gezogen werden, dass

- Ein Paket eines bestimmten 5-Tupels (src/dst IP, src/dst Port, Protokoll) kommt an der Firewall an der erwarteten Schnittstelle (INSIDE) an.
- Ein Paket verlässt die Firewall nicht auf der erwarteten Schnittstelle (AUSSEN).

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Überprüfen der Ablaufverfolgung eines emulierten Pakets

Verwenden Sie das Tool zur Paketverfolgung, um festzustellen, wie ein Paket von der Firewall behandelt werden soll. Wenn das Paket von der Firewall-Zugriffsrichtlinie verworfen wird, sieht die Ablaufverfolgung des emulierten Pakets ähnlich wie diese Ausgabe aus:

```

<#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
```

Maßnahme 2: Überprüfen Sie die Spuren von aktiven Paketen.

Aktivieren Sie die Paketverfolgung, um zu überprüfen, wie die echten TCP-SYN-Pakete von der Firewall verarbeitet werden. Standardmäßig werden nur die ersten 50 eingehenden Pakete verfolgt:

```
<#root>
firepower#
capture CAPI trace
```

Löschen Sie den Erfassungspuffer:

```
<#root>
firepower#
clear capture /all
```

Falls das Paket von der Firewall-Zugriffsrichtlinie verworfen wird, sieht die Ablaufverfolgung ähnlich wie diese Ausgabe aus:

```
<#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 <ms
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

Maßnahme 3: FTD-Lina-Protokolle überprüfen.

Um Syslog auf FTD über FMC zu konfigurieren, lesen Sie dieses Dokument:

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

Es wird dringend empfohlen, einen externen Syslog-Server für FTD-Lina-Protokolle zu konfigurieren. Wenn kein entfernter Syslog-Server konfiguriert ist, aktivieren Sie während der Fehlerbehebung lokale Pufferprotokolle in der Firewall. Die in diesem Beispiel gezeigte Protokollkonfiguration ist ein guter Ausgangspunkt:

```
<#root>
```

```
firepower#
```

```
show run logging
```

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

Stellen Sie den Terminal-Pager auf 24 Leitungen ein, um den Terminal-Pager zu steuern:

```
<#root>
```

```
firepower#
```


Löschen Sie den Erfassungspuffer:

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

Testen Sie die Verbindung, und überprüfen Sie die Protokolle mit einem Parserfilter. In diesem Beispiel werden die Pakete von der Firewall-Zugriffsrichtlinie verworfen:

```
<#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 b
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 b
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 b
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 b
```

Maßnahme 4: Überprüfen Sie die Firewall-ASP-Drops.

Wenn Sie vermuten, dass das Paket von der Firewall verworfen wird, können Sie die Zähler aller Pakete sehen, die von der Firewall auf Softwareebene verworfen wurden:

```
<#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
```

Sie können Aufnahmen aktivieren, um alle ASP-Verwerfungen auf Softwareebene anzuzeigen:

```
<#root>
```

```
firepower#
```

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

Tip: Wenn Sie sich nicht für den Paketinhalt interessieren, können Sie nur die Paket-Header erfassen (nur Header-Option). Dadurch können Sie viel mehr Pakete im Capture-Puffer erfassen. Zusätzlich können Sie die Größe des Capture-Puffers (standardmäßig 500 KB) auf einen Wert von bis zu 32 MB (Pufferoption) erhöhen. Ab FTD Version 6.3 können Sie mit der Dateigrößenoption eine Erfassungsdatei von bis zu 10 GByte konfigurieren. In diesem Fall können Sie den Inhalt der Aufnahme nur im pcap-Format sehen.

Um den Inhalt der Erfassung zu überprüfen, können Sie die Suche mithilfe eines Filters eingrenzen:

```
<#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 1
19: 07:51:58.074291 192.168.0.100.12411 > 10.10.1.100.80: S 2006489005:2006489005(0) win 8192 <mss 1
26: 07:52:00.830370 192.168.0.100.12410 > 10.10.1.100.80: S 1870382552:1870382552(0) win 8192 <mss 1
29: 07:52:01.080394 192.168.0.100.12411 > 10.10.1.100.80: S 2006489005:2006489005(0) win 8192 <mss 1
45: 07:52:06.824282 192.168.0.100.12410 > 10.10.1.100.80: S 1870382552:1870382552(0) win 8192 <mss 1
46: 07:52:07.074230 192.168.0.100.12411 > 10.10.1.100.80: S 2006489005:2006489005(0) win 8192 <mss 1
```

Da die Pakete in diesem Fall bereits auf Schnittstellenebene verfolgt werden, wird der Grund für den Verfall in der ASP-Erfassung nicht erwähnt. Denken Sie daran, dass ein Paket nur an einem Ort verfolgt werden kann (Eingangsschnittstelle oder ASP-Drop). In diesem Fall wird empfohlen, mehrere ASP-Drops durchzuführen und einen bestimmten ASP-Dropgrund festzulegen. Dies ist ein empfohlener Ansatz:

1. Löschen Sie die aktuellen ASP-Zähler zum Ablegen:

```
<#root>
```

```
firepower#
```

```
clear asp drop
```

2. Senden Sie den Fluss, den Sie bei der Fehlerbehebung verwenden, über die Firewall (führen Sie einen Test aus).

3. Überprüfen Sie erneut die ASP-Zähler für das Ablegen, und notieren Sie die erhöhten Zähler.

```
<#root>
```

```
firepower#
```

```
show asp drop
```

```
Frame drop:
```

```
No route to host (
```

```

no-route
)
Flow is denied by configured rule (
acl-drop
)
71
234

```

4. Aktivieren Sie ASP-Erfassung(en) für die bestimmten erkannten Drops:

```

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

```

5. Senden Sie den Fluss, den Sie bei der Fehlerbehebung verwenden, über die Firewall (führen Sie einen Test aus).

6. Überprüfen Sie die ASP-Aufnahmen. In diesem Fall wurden die Pakete aufgrund einer fehlenden Route verworfen:

```

<#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 1
95: 07:53:52.632337 192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss 1
101: 07:53:55.375392 192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss 1
102: 07:53:55.626386 192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss 1
116: 07:54:01.376231 192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss 1
117: 07:54:01.626310 192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss 1

```

Maßnahme 5: Überprüfen Sie die FTD Lina-Verbindungstabelle.

Es kann Fälle geben, in denen Sie erwarten, dass das Paket die Schnittstelle 'X' verlässt, aber aus welchen Gründen auch immer, es verlässt die Schnittstelle 'Y'. Die Bestimmung der Firewall-Ausgangsschnittstelle basiert auf der folgenden Reihenfolge:

1. Suche nach Verbindung wurde eingerichtet
2. NAT-Suche (Network Address Translation) - Die UN-NAT-Phase (Ziel-NAT) hat Vorrang vor der PBR- und Routensuche.
3. Richtlinienbasiertes Routing
4. Routingtabellen-Suche

So überprüfen Sie die FTD-Verbindungstabelle:

```
<#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

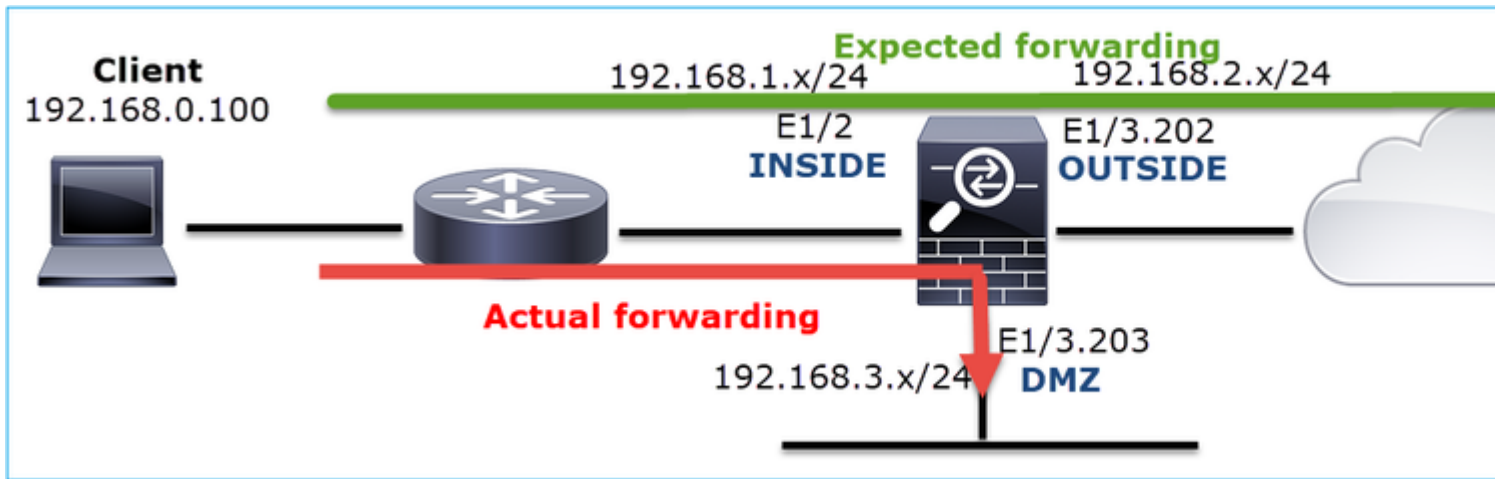
TCP
DMZ
    10.10.1.100:
80

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
```

Wichtigste Punkte:

- Basierend auf den Flaggen (Aa) ist die Verbindung embryonal (halb geöffnet - nur TCP SYN wurde von der Firewall gesehen).
- Auf Basis der Quell-/Ziel-Ports ist die Eingangsschnittstelle INSIDE, und die Ausgangsschnittstelle ist DMZ.

Dies kann im Bild hier visualisiert werden:



Hinweis: Da alle FTD-Schnittstellen die Sicherheitsstufe 0 haben, basiert die Schnittstellenreihenfolge in der **angezeigten** Verbindungsausgabe auf der Schnittstellenummer. Insbesondere wird die Schnittstelle mit höherer vpif-num (Virtual Platform Interface Number) als inside und die Schnittstelle mit niedrigerer vpif-num als outside ausgewählt. Der Wert für interface vpif wird mit dem Befehl **show interface detail** angezeigt. Zugehörige Erweiterung, Cisco Bug-ID [CSCvi15290](https://www.cisco.com/cisco/webbugtools/bugdetail.do?bugs=CSCvi15290) ENH: FTD zeigt die Verbindungsrichtung in der FTD-"show conn"-Ausgabe an

```
<#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
```

Hinweis: Ab Firepower Software 6.5, ASA Version 9.13.x liefern die Befehlsausgaben show conn long und show conn detail Informationen über den Verbindungsinitiator und den Responder

Ausgabe 1:

```
<#root>
firepower#
show conn long
...
```

TCP OUTSIDE: 192.168.2.200/80 (192.168.2.200/80) INSIDE: 192.168.1.100/46050 (192.168.1.100/46050), flags

Initiator: 192.168.1.100, Responder: 192.168.2.200

Connection lookup keyid: 228982375

Ausgabe 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

Zusätzlich zeigt **show conn long** die NATed-IPs in einer Klammer an, wenn es sich um eine Network Address Translation handelt:

```
<#root>
```

```
firepower#
```

```
show conn long
```

```
...
```

```
TCP OUTSIDE: 192.168.2.222/80 (192.168.2.222/80) INSIDE: 192.168.1.100/34792 (192.168.2.150/34792), flags  
  Initiator: 192.168.1.100, Responder: 192.168.2.222  
  Connection lookup keyid: 262895
```

Maßnahme 6. Überprüfen Sie den ARP-Cache (Address Resolution Protocol) der Firewall.

Wenn die Firewall den nächsten Hop nicht auflösen kann, verwirft die Firewall unbeaufsichtigt das ursprüngliche Paket (in diesem Fall TCP SYN) und sendet kontinuierlich ARP-Anforderungen, bis der nächste Hop aufgelöst ist.

Um den ARP-Cache der Firewall anzuzeigen, verwenden Sie den folgenden Befehl:

```
<#root>
```

```
firepower#
```

```
show arp
```

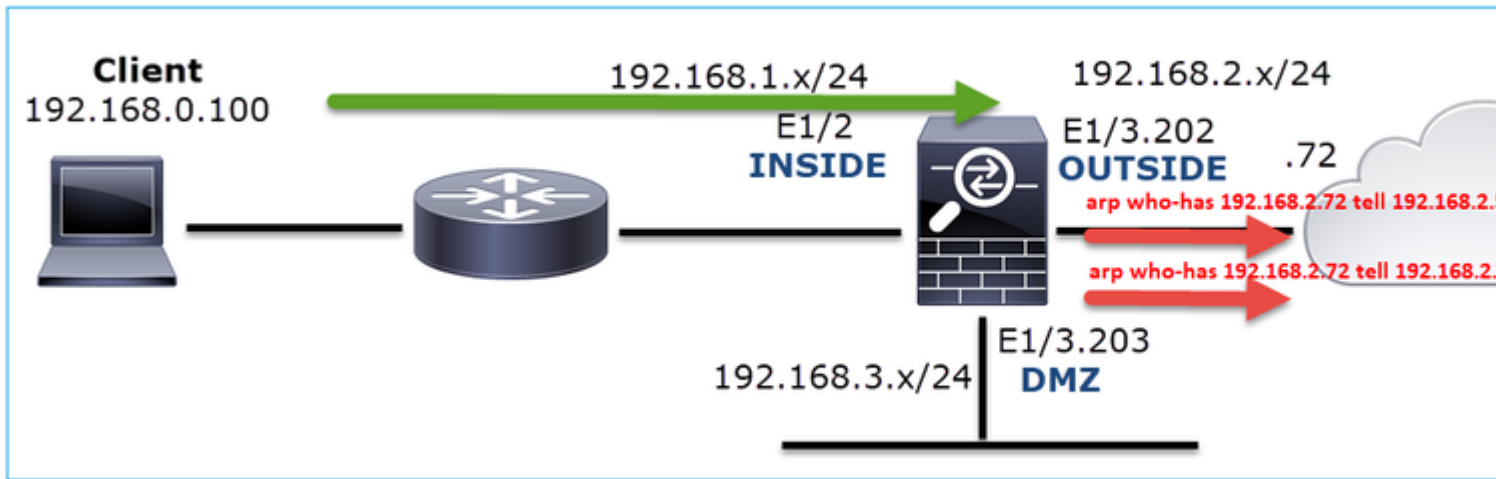
Um zu überprüfen, ob es nicht aufgelöste Hosts gibt, können Sie den folgenden Befehl verwenden:

```
<#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
```

Wenn Sie den ARP-Vorgang weiter überprüfen möchten, können Sie eine ARP-spezifische Erfassung aktivieren:

```
<#root>
firepower#
capture ARP ethernet-type arp interface OUTSIDE
firepower#
show capture ARP
...
 4: 07:15:16.877914      802.1Q vlan#202 P0 arp
who-has 192.168.2.72 tell 192.168.2.50
 5: 07:15:18.020033      802.1Q vlan#202 P0 arp who-has 192.168.2.72 tell 192.168.2.50
```

In dieser Ausgabe versucht die Firewall (192.168.2.50), den nächsten Hop (192.168.2.72) aufzulösen, es gibt jedoch keine ARP-Antwort.



Die Ausgabe hier zeigt ein funktionelles Szenario mit der richtigen ARP-Auflösung:

```
<#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
```

Falls kein ARP-Eintrag vorhanden ist, zeigt die Ablaufverfolgung eines aktiven TCP-SYN-Pakets Folgendes an:

```
<#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

â€¦

Phase: 14

Type: FLOW-CREATION

Subtype:

Result: ALLOW

Config:

Additional Information:

New flow created with id 4814, packet dispatched to next module

â€¦

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

Wie in der Ausgabe zu sehen ist, zeigt die Ablaufverfolgung "**Action: allow**" an, selbst wenn der nächste Hop nicht erreichbar ist und das Paket stumm von der Firewall verworfen wird! In diesem Fall muss auch das Paket-Tracer-Tool überprüft werden, da es eine genauere Ausgabe ermöglicht:

<#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
â€¦

Phase: 14
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 4816, packet dispatched to next module
â€¦

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)

Bei den aktuellen ASA/Firepower-Versionen wurde die vorherige Meldung optimiert, um:

<#root>

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

Check ARP table (show arp) has entry for nexthop

., Drop-location: f

Mögliche Ursachen und empfohlene Aktionen - Zusammenfassung

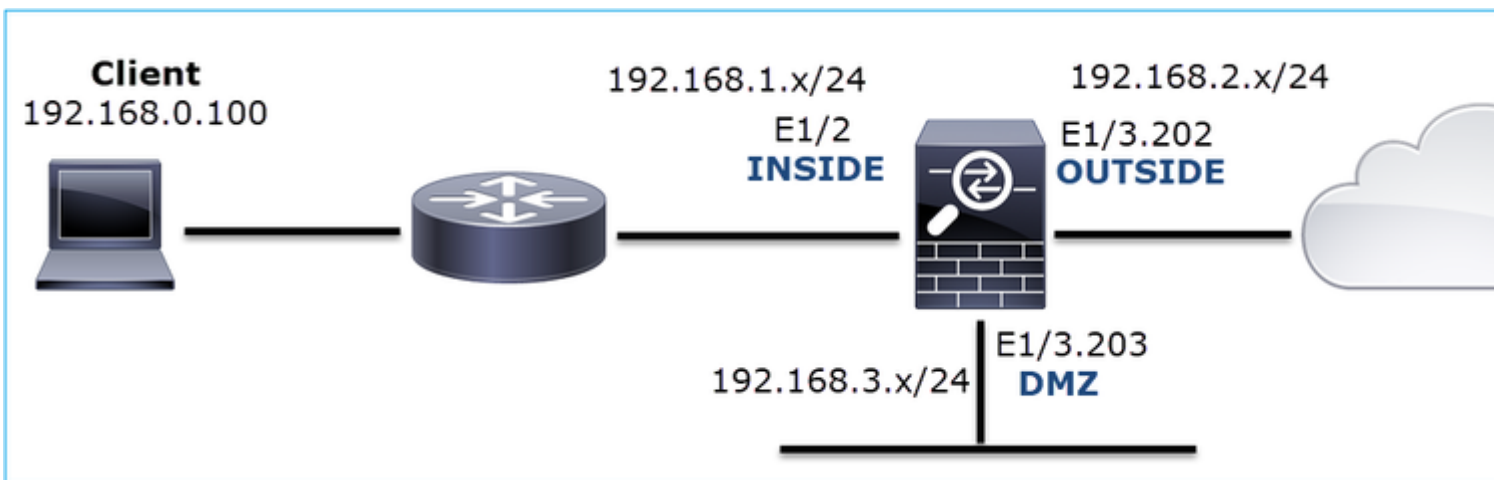
Wenn Sie nur ein TCP-SYN-Paket an den Eingangsschnittstellen sehen, aber kein TCP-SYN-Paket von der erwarteten Ausgangsschnittstelle gesendet wird, gibt es einige mögliche Ursachen:

Mögliche Ursache	Empfohlene Maßnahmen
Das Paket wird von der Firewall-Zugriffsrichtlinie verworfen.	<ul style="list-style-type: none">• Verwenden Sie Packet-Tracer oder capture w/trace, um zu sehen, wie die Firewall mit dem Paket umgeht.• Firewall-Protokolle überprüfen.• Überprüfen Sie die Firewall-ASP-Drops (show asp drop oder capture type asp-drop).• FMC-Verbindungsereignisse überprüfen. Dabei wird davon ausgegangen, dass für die Regel die Protokollierung aktiviert ist.
Der Erfassungsfiler ist falsch.	<ul style="list-style-type: none">• Verwenden Sie die Paketverfolgung oder Erfassung mit Trace, um festzustellen, ob eine NAT-Übersetzung vorliegt, die die Quell- oder Ziel-IP-Adresse ändert. Passen Sie in diesem Fall den Erfassungsfiler an.• show conn long zeigt die NATed-IPs an.
Das Paket wird an eine andere Ausgangsschnittstelle gesendet.	<ul style="list-style-type: none">• Verwenden Sie die Paketverfolgung oder Erfassung mit Ablaufverfolgung, um zu sehen, wie die Firewall mit dem Paket umgeht. Denken Sie an die Reihenfolge der Vorgänge, die die Ermittlung der Ausgangsschnittstelle, die aktuelle Verbindung, UN-NAT, PBR und die Suche in der Routing-Tabelle betreffen.• Firewall-Protokolle überprüfen.• Überprüfen Sie die Firewall-Verbindungstabelle (show conn). <p>Wenn das Paket an eine falsche Schnittstelle gesendet wird, weil es mit einer aktuellen Verbindung übereinstimmt, verwenden Sie den Befehl clear conn address, und geben Sie das 5-Tupel der Verbindung an, die gelöscht werden soll.</p>

<p>Es gibt keine Route zum Ziel.</p>	<ul style="list-style-type: none"> • Verwenden Sie Packet-Tracer oder capture w/trace, um zu sehen, wie die Firewall mit dem Paket umgeht. • Überprüfen Sie die Firewall-ASP-Drops (show asp drop) auf Gründe für ein nicht rotierendes Dropdown.
<p>Es gibt keinen ARP-Eintrag auf der Ausgangsschnittstelle.</p>	<ul style="list-style-type: none"> • Überprüfen Sie den Firewall-ARP-Cache (show arp). • Verwenden Sie die Paketverfolgung, um festzustellen, ob eine gültige Adjacency vorhanden ist.
<p>Die Ausgangsschnittstelle ist ausgefallen.</p>	<p>Überprüfen Sie die Ausgabe des Befehls show interface ip brief auf der Firewall, und überprüfen Sie den Schnittstellenstatus.</p>

Fall 2: TCP SYN vom Client, TCP RST vom Server

Dieses Bild zeigt die Topologie:



Problembeschreibung: HTTP funktioniert nicht

Betroffener Datenfluss:

Quelle IP: 192.168.0.100

Ziel-IP: 10.10.1.100

Protokoll: TCP 80

Erfassungsanalyse

Aktivieren Sie Aufnahmen auf der FTD LINA-Engine.

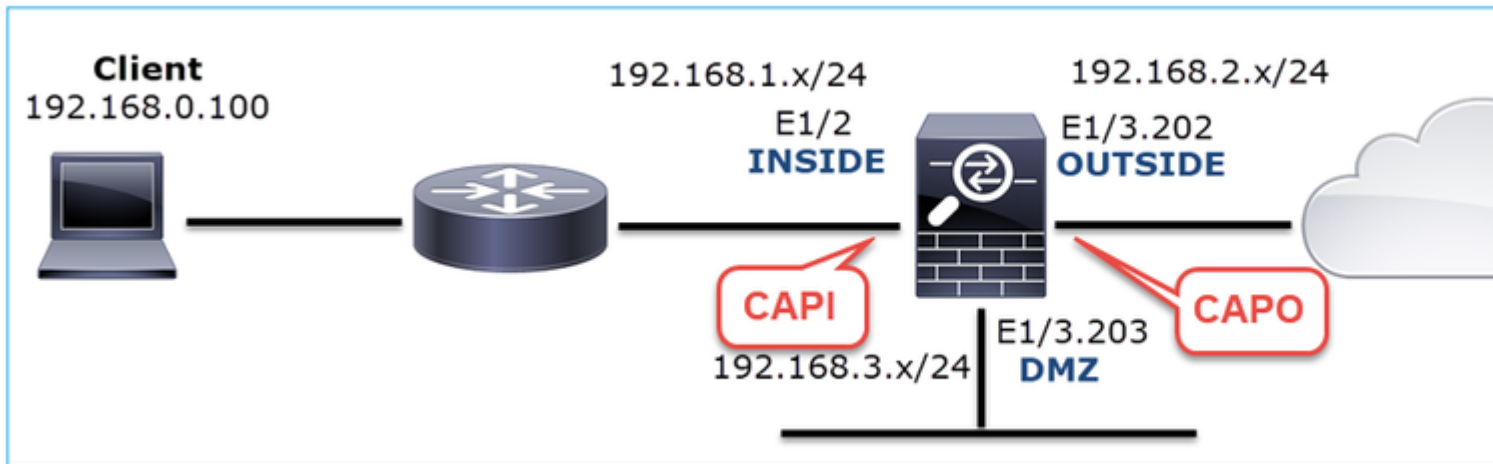
```
<#root>
```

```
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
```



Erfassungen - Nicht-funktionales Szenario:

Aus der Geräte-CLI sehen die Aufnahmen wie folgt aus:

```
<#root>
```

```
firepower#
```

```
show capture
```

```
capture CAPI type raw-data trace interface INSIDE [Capturing -
```

```
834 bytes
```

```
]
```

```
match ip host 192.168.0.100 host 10.10.1.100
```

```
capture CAPO type raw-data interface OUTSIDE [Capturing -
```

```
878 bytes
```

```
]
```

```
match ip host 192.168.0.100 host 10.10.1.100
```

CAPI-Inhalt:

```
<#root>
```

```
firepower#
```

show capture CAPI

1: 05:20:36.654217 192.168.0.100.22195 > 10.10.1.100.80:

S

1397289928:1397289928(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>

2: 05:20:36.904311 192.168.0.100.22196 > 10.10.1.100.80:

S

2171673258:2171673258(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>

3: 05:20:36.905043 10.10.1.100.80 > 192.168.0.100.22196:

R

1850052503:1850052503(0) ack 2171673259 win 0

4: 05:20:37.414132 192.168.0.100.22196 > 10.10.1.100.80:

S

2171673258:2171673258(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>

5: 05:20:37.414803 10.10.1.100.80 > 192.168.0.100.22196:

R

31997177:31997177(0) ack 2171673259 win 0

6: 05:20:37.914183 192.168.0.100.22196 > 10.10.1.100.80:

S

2171673258:2171673258(0) win 8192 <mss 1460,nop,nop,sackOK>

...

CAPO-Inhalt:

<#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>
```

Dieses Bild zeigt die Erfassung von CAPI in Wireshark.

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

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

Wichtigste Punkte:

1. Die Quelle sendet ein TCP SYN-Paket.
2. Eine TCP-RST wird an die Quelle gesendet.
3. Die Quelle überträgt die TCP-SYN-Pakete erneut.
4. Die MAC-Adressen sind korrekt (bei eingehenden Paketen gehört die Quell-MAC-Adresse zum Downstream-Router, die Ziel-MAC-Adresse zur INSIDE-Schnittstelle der Firewall).

Dieses Bild zeigt die Aufnahme von CAPO in Wireshark:

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

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
 > Ethernet II, Src: Cisco_f6:1d:8e (00:be:75:f6:1d:8e), Dst: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8)
 > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
 > Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Seq: 0, Len: 0

Wichtigste Punkte:

1. Die Quelle sendet ein TCP SYN-Paket.
2. Eine TCP-RST kommt über die OUTSIDE-Schnittstelle an.
3. Die Quelle überträgt die TCP-SYN-Pakete erneut.
4. Die MAC-Adressen sind richtig (bei Ausgangspaketen ist die Firewall AUSSERHALB die Quell-MAC, der Upstream-Router die Ziel-MAC).

Auf der Grundlage der beiden Aufzeichnungen kann der Schluss gezogen werden, dass

- Der TCP-3-Wege-Handshake zwischen Client und Server wird nicht abgeschlossen
- Es gibt eine TCP-RST, die an der Firewall-Ausgangsschnittstelle eingeht.
- Die Firewall kommuniziert mit den entsprechenden Upstream- und Downstream-Geräten (basierend auf den MAC-Adressen).

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Überprüfen Sie die Quell-MAC-Adresse, die die TCP-RST sendet.

Überprüfen Sie, ob die Ziel-MAC im TCP-SYN-Paket mit der Quell-MAC im TCP-RST-Paket übereinstimmt.

The image displays two screenshots of the Wireshark network protocol analyzer, showing a sequence of events in a packet capture file named 'CAPO_RST_SERVER.pcap'.

Top Screenshot (Frame 2): Shows a TCP SYN packet (No. 2) sent from source IP 192.168.0.100 to destination IP 10.10.1.100. The source MAC address is Cisco_f6:1d:8e (00:be:75:f6:1d:8e) and the destination MAC address is Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8).

Bottom Screenshot (Frame 3): Shows a TCP RST, ACK packet (No. 3) sent from source IP 10.10.1.100 to destination IP 192.168.0.100. The source MAC address is Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8) and the destination MAC address is Cisco_f6:1d:8e (00:be:75:f6:1d:8e).

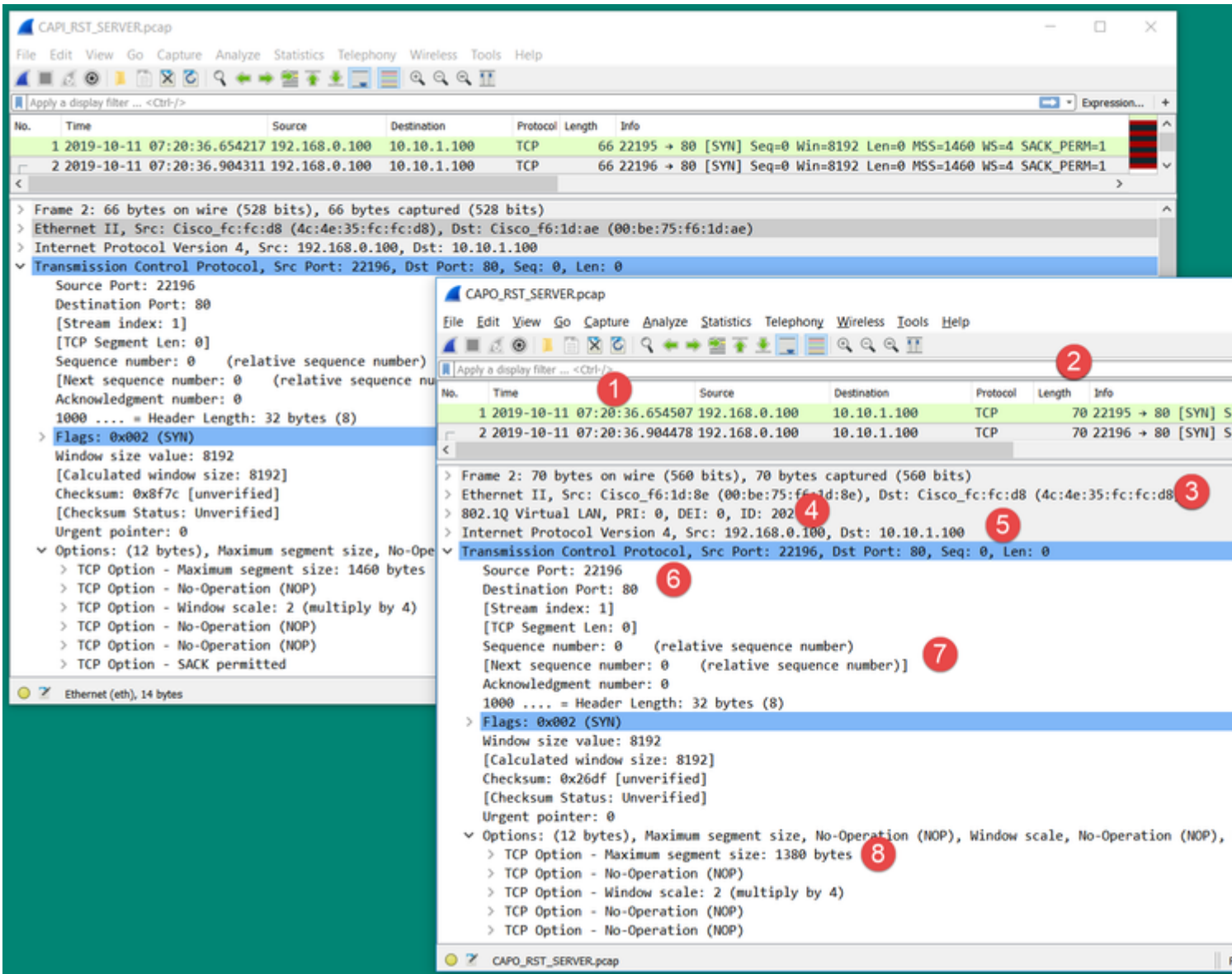
Orange and green boxes highlight the MAC addresses in both screenshots. Two arrows, one orange and one green, cross each other, pointing from the source MAC of the SYN packet in the top frame to the source MAC of the RST packet in the bottom frame, illustrating that they do not match.

Diese Prüfung hat zum Ziel, 2 Dinge zu bestätigen:

- Stellen Sie sicher, dass kein asymmetrischer Fluss vorhanden ist.
- Überprüfen Sie, ob die MAC-Adresse zum erwarteten Upstream-Gerät gehört.

Maßnahme 2: Vergleich von ein- und ausgehenden Paketen

Vergleichen Sie visuell die beiden Pakete in Wireshark, um sicherzustellen, dass die Firewall die Pakete nicht verändert/beschädigt. Einige erwartete Unterschiede werden hervorgehoben.



Wichtigste Punkte:

1. Die Zeitstempel sind unterschiedlich. Auf der anderen Seite muss der Unterschied klein und vernünftig sein. Dies hängt von den Funktionen und Richtlinienprüfungen ab, die auf das Paket angewendet werden, sowie von der Last auf dem Gerät.
2. Die Länge der Pakete kann sich insbesondere dann unterscheiden, wenn ein dot1Q-Header von der Firewall nur auf einer Seite hinzugefügt/entfernt wird.
3. Die MAC-Adressen sind unterschiedlich.
4. Ein dot1Q-Header kann vorhanden sein, wenn die Erfassung auf einer Schnittstelle durchgeführt wurde.
5. Die IP-Adresse(n) ist/sind unterschiedlich, wenn NAT oder Port Address Translation (PAT) auf das

Paket angewendet wird.

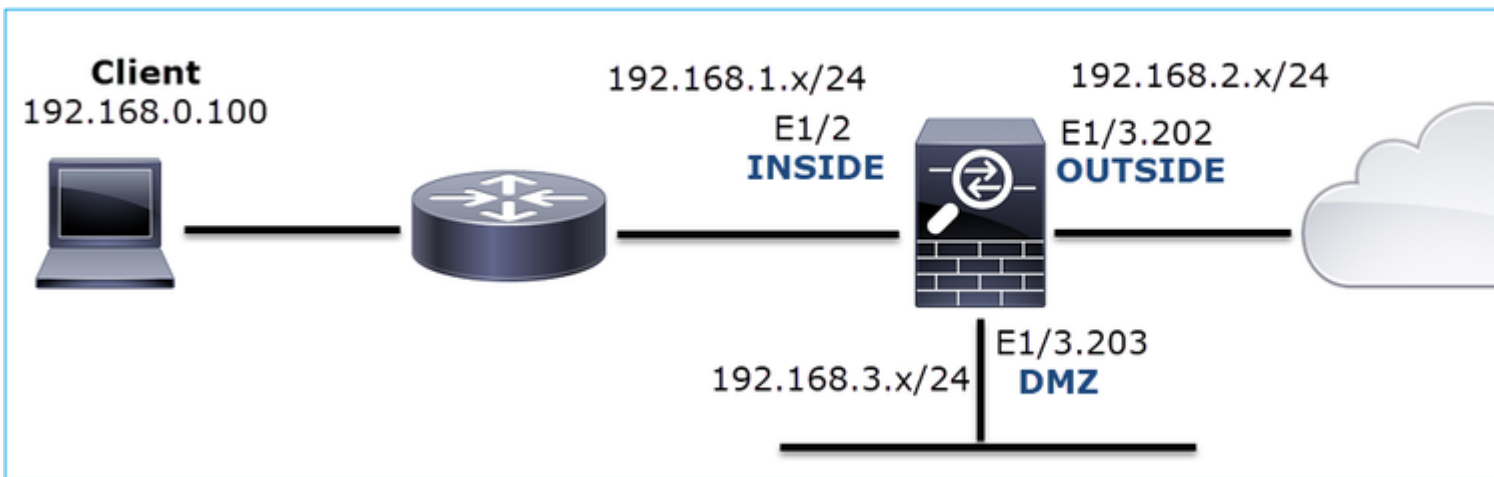
6. Die Quell- oder Ziel-Ports unterscheiden sich, wenn NAT oder PAT auf das Paket angewendet wird.
7. Wenn Sie die Wireshark **Relative Sequence Number**-Option deaktivieren, stellen Sie fest, dass die TCP-Sequenznummern/Bestätigungsnummern durch die Firewall aufgrund der ISN-Randomisierung (Initial Sequence Number) geändert werden.
8. Einige TCP-Optionen können überschrieben werden. Beispielsweise ändert die Firewall standardmäßig die maximale TCP-Segmentgröße (Maximum Segment Size, MSS) auf 1380, um eine Paketfragmentierung im Transportpfad zu vermeiden.

Maßnahme 3: Nehmen Sie eine Aufnahme am Ziel.

Wenn möglich, machen Sie eine Erfassung am Ziel selbst. Wenn dies nicht möglich ist, nehmen Sie eine Erfassung so nahe wie möglich am Ziel vor. Ziel hierbei ist es zu überprüfen, wer die TCP-RST sendet (handelt es sich um den Zielservers oder ein anderes Gerät im Pfad?).

Fall 3: TCP 3-Wege-Handshake + RST von einem Endgerät

Dieses Bild zeigt die Topologie:



Problembeschreibung: HTTP funktioniert nicht

Betroffener Datenfluss:

Quelle IP: 192.168.0.100

Ziel-IP: 10.10.1.100

Protokoll: TCP 80

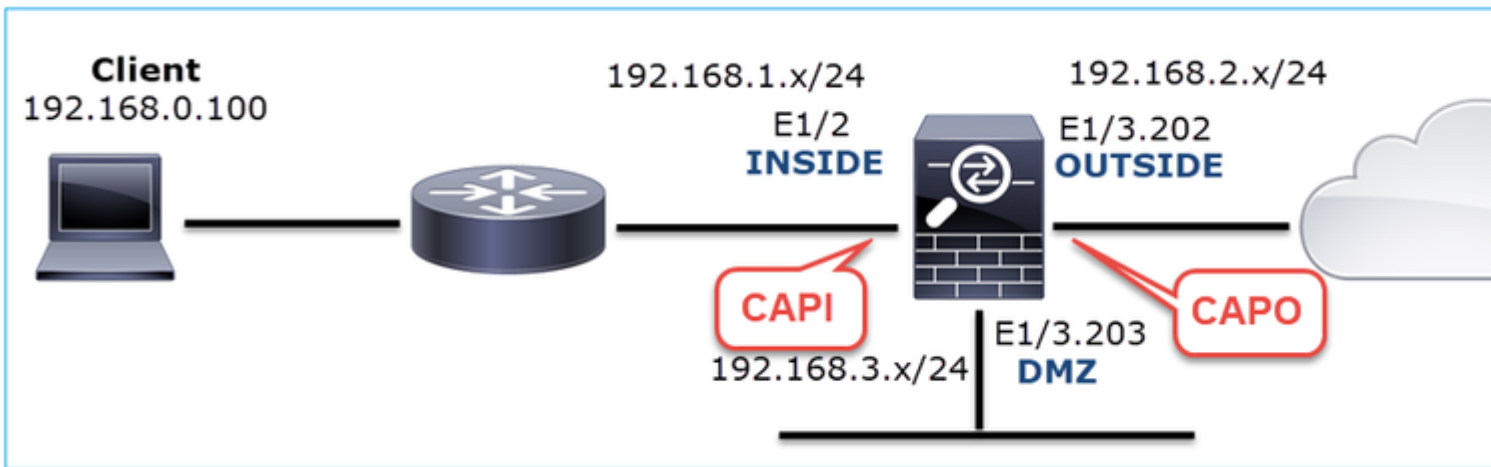
Erfassungsanalyse

Aktivieren Sie Aufnahmen auf der FTD LINA-Engine.

```
<#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
```



Erfassungen - Nicht-funktionales Szenario:

Es gibt verschiedene Möglichkeiten, wie dieses Problem in Aufnahmen manifestiert werden kann.

3.1 - TCP-3-Wege-Handshake + verzögerte RST vom Client

Sowohl die Firewall als auch der CAPI enthalten die gleichen Pakete, wie im Bild gezeigt.

No.	Time	Source	Destination	Protocol	Length	Info
2	2019-10-13 17:06:27.874085	192.168.0.100	10.10.1.100	TCP	66	48295 → 80 [SYN] Seq=179631561 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	2019-10-13 17:06:27.874741	10.10.1.100	192.168.0.100	TCP	66	80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380
4	2019-10-13 17:06:27.875183	192.168.0.100	10.10.1.100	TCP	54	48295 → 80 [ACK] Seq=179631562 Ack=3838911938 Win=66240 Len=0
8	2019-10-13 17:06:30.882537	10.10.1.100	192.168.0.100	TCP	66	[TCP Retransmission] 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380
9	2019-10-13 17:06:30.883056	192.168.0.100	10.10.1.100	TCP	66	[TCP Previous segment not captured] 48295 → 80 [ACK] Seq=179631962 Ack=3838911938 Win=66240 Len=0
13	2019-10-13 17:06:36.889022	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380
14	2019-10-13 17:06:36.889526	192.168.0.100	10.10.1.100	TCP	66	[TCP Dup ACK 4#1] 48295 → 80 [ACK] Seq=179631962 Ack=3838911938 Win=66240 Len=0
17	2019-10-13 17:06:47.943631	192.168.0.100	10.10.1.100	TCP	54	48295 → 80 [RST, ACK] Seq=179631962 Ack=3838911938 Win=0 Len=0

Wichtigste Punkte:

1. Der TCP 3-Wege Handshake geht durch die Firewall.
2. Der Server sendet erneut SYN/ACK.
3. Der Client sendet die ACK erneut.
4. Nach ~20 Sekunden gibt der Client auf und sendet eine TCP-RST.

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Nehmen Sie Aufnahmen so nah wie möglich an den beiden Endpunkten vor.

Die Firewall-Erfassungen zeigen an, dass die Client-ACK nicht vom Server verarbeitet wurde. Dies beruht auf folgenden Tatsachen:

- Der Server sendet erneut SYN/ACK.
- Der Client sendet die ACK erneut.

- Der Client sendet eine TCP-RST oder eine FIN/ACK-Nachricht vor allen Daten.

Erfassung auf dem Server zeigt das Problem. Der Client ACK vom TCP 3-Wege-Handshake kam nie an:

26	7.636612	192.168.0.100	10.10.1.100	TCP	66	55324→80 [SYN] Seq=43320132
29	7.637571	10.10.1.100	192.168.0.100	TCP	66	80→55324 [SYN, ACK] Seq=406
30	7.930152	192.168.0.100	10.10.1.100	TCP	66	55325→80 [SYN] Seq=36619749
31	7.930221	10.10.1.100	192.168.0.100	TCP	66	80→55325 [SYN, ACK] Seq=215
41	10.629868	192.168.0.100	10.10.1.100	TCP	66	[TCP Spurious Retransmission]
42	10.633208	10.10.1.100	192.168.0.100	TCP	66	[TCP Retransmission] 80→553
44	10.945178	10.10.1.100	192.168.0.100	TCP	66	[TCP Retransmission] 80→553
60	16.636255	192.168.0.100	10.10.1.100	TCP	62	[TCP Spurious Retransmission]
61	16.639145	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80→553
62	16.951195	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80→553

3.2 - TCP-3-Wege-Handshake + verzögerte FIN/ACK vom Client + verzögerte RST vom Server

Sowohl die Firewall als auch der CAPI enthalten die gleichen Pakete, wie im Bild gezeigt.

25	2019-10-13 17:07:06.853334	192.168.0.100	10.10.1.100	TCP	66	48299 → 80 [SYN] Seq=3239914002 Win=8192 Len=0 MSS=1460 WS=4 S
29	2019-10-13 17:07:09.852922	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 48299 → 80 [SYN] Seq=3239914002 Win=8192
30	2019-10-13 17:07:09.854844	10.10.1.100	192.168.0.100	TCP	66	80 → 48299 [SYN, ACK] Seq=808763519 Ack=3239914003 Win=8192 L
31	2019-10-13 17:07:09.855287	192.168.0.100	10.10.1.100	TCP	54	48299 → 80 [ACK] Seq=3239914003 Ack=808763520 Win=66240 Len=0
34	2019-10-13 17:07:14.856996	192.168.0.100	10.10.1.100	TCP	54	48299 → 80 [FIN, ACK] Seq=3239914003 Ack=808763520 Win=66240 L
35	2019-10-13 17:07:15.861451	10.10.1.100	192.168.0.100	TCP	62	[TCP Retransmission] 80 → 48299 [SYN, ACK] Seq=808763519 Ack=3
36	2019-10-13 17:07:15.861970	192.168.0.100	10.10.1.100	TCP	66	[TCP Dup ACK 31#1] 48299 → 80 [ACK] Seq=3239914004 Ack=808763
39	2019-10-13 17:07:17.854051	192.168.0.100	10.10.1.100	TCP	54	[TCP Retransmission] 48299 → 80 [FIN, ACK] Seq=3239914003 Ack=
40	2019-10-13 17:07:23.855012	192.168.0.100	10.10.1.100	TCP	54	[TCP Retransmission] 48299 → 80 [FIN, ACK] Seq=3239914003 Ack=
46	2019-10-13 17:07:27.858949	10.10.1.100	192.168.0.100	TCP	54	80 → 48299 [RST] Seq=808763520 Win=0 Len=0

Wichtigste Punkte:

1. Der TCP 3-Wege Handshake geht durch die Firewall.
2. Nach ca. 5 Sekunden sendet der Client eine FIN/ACK.
3. Nach ~20 Sekunden gibt der Server auf und sendet eine TCP-RST.

Auf Basis dieser Erfassung kann geschlossen werden, dass es zwar einen TCP 3-Wege-Handshake durch die Firewall gibt, es aber so aussieht, als würde er nie an einem Endpunkt abgeschlossen (die Neuübertragungen deuten darauf hin).

Empfohlene Maßnahmen

Wie in Fall 3.1

3.3 - TCP-3-Wege-Handshake + verzögerte RST vom Client

Sowohl die Firewall als auch der CAPI enthalten die gleichen Pakete, wie im Bild gezeigt.

No.	Time	Source	Destination	Protocol	Length	Info
129	2019-10-13 17:09:20.513355	192.168.0.100	10.10.1.100	TCP	66	48355 → 80 [SYN] Seq=2581697538 Wi
130	2019-10-13 17:09:20.514011	10.10.1.100	192.168.0.100	TCP	66	80 → 48355 [SYN, ACK] Seq=16330186
131	2019-10-13 17:09:20.514438	192.168.0.100	10.10.1.100	TCP	54	48355 → 80 [ACK] Seq=2581697539 Ac
132	2019-10-13 17:09:39.473089	192.168.0.100	10.10.1.100	TCP	54	48355 → 80 [RST, ACK] Seq=25816979

Wichtigste Punkte:

1. Der TCP 3-Wege Handshake geht durch die Firewall.
2. Nach ~20 Sekunden gibt der Client auf und sendet eine TCP-RST.

Auf der Grundlage dieser Aufzeichnungen kann der Schluss gezogen werden, dass:

- Nach 5-20 Sekunden gibt ein Endpunkt auf und beschließt, die Verbindung zu beenden.

Empfohlene Maßnahmen

Wie in Fall 3.1

3.4 - TCP-3-Wege-Handshake + sofortige RST vom Server

Sowohl die Firewall als auch der CAPI enthalten diese Pakete, wie im Bild gezeigt.

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=
27	2019-10-13 17:07:07.105112	10.10.1.100	192.168.0.100	TCP	66	80 → 48300 [SYN, ACK] Seq=3757137497
28	2019-10-13 17:07:07.105554	192.168.0.100	10.10.1.100	TCP	54	48300 → 80 [ACK] Seq=2563435280 Ack=
41	2019-10-13 17:07:07.106325	10.10.1.100	192.168.0.100	TCP	54	80 → 48300 [RST] Seq=2563435280 Win=

Wichtigste Punkte:

1. Der TCP 3-Wege Handshake geht durch die Firewall.
2. Ein paar Millisekunden nach dem ACK-Paket erfolgt eine TCP-RST vom Server.

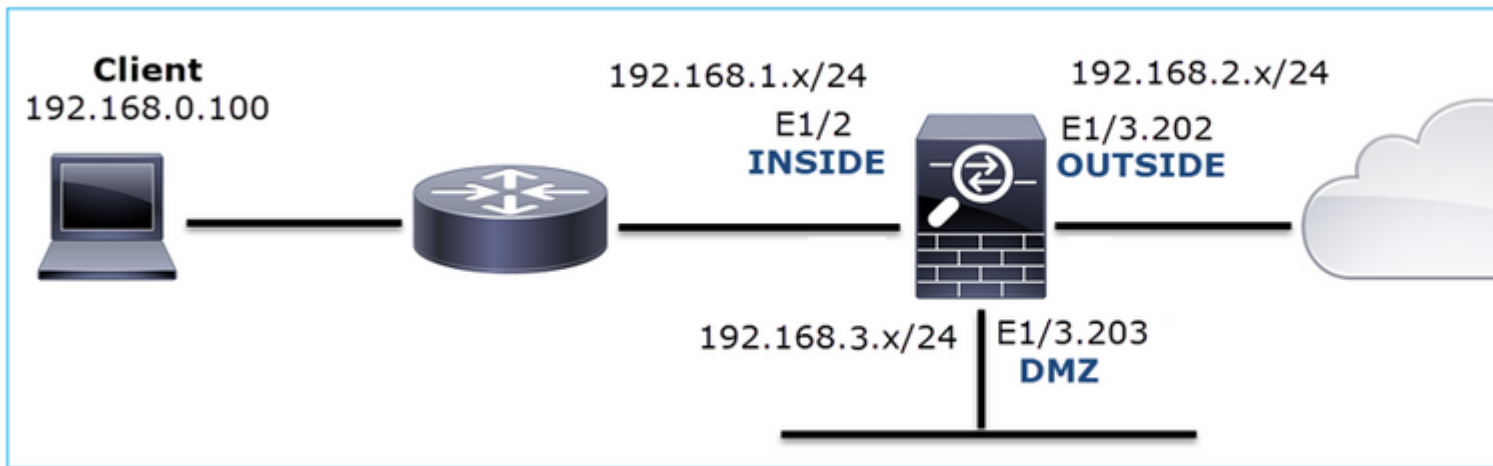
Empfohlene Maßnahmen

Aktion: Nehmen Sie Aufnahmen so nah wie möglich am Server vor.

Eine sofortige TCP-RST vom Server kann auf einen fehlerhaften Server oder ein fehlerhaftes Gerät im Pfad hinweisen, von dem die TCP-RST gesendet wird. Erfassen Sie den Server selbst, und bestimmen Sie die Quelle der TCP-RST.

Fall 4: TCP RST vom Client

Dieses Bild zeigt die Topologie:



Problembeschreibung: HTTP funktioniert nicht.

Betroffener Datenfluss:

Quelle IP: 192.168.0.100

Ziel-IP: 10.10.1.100

Protokoll: TCP 80

Erfassungsanalyse

Aktivieren Sie Aufnahmen auf FTD LINA-Engine.

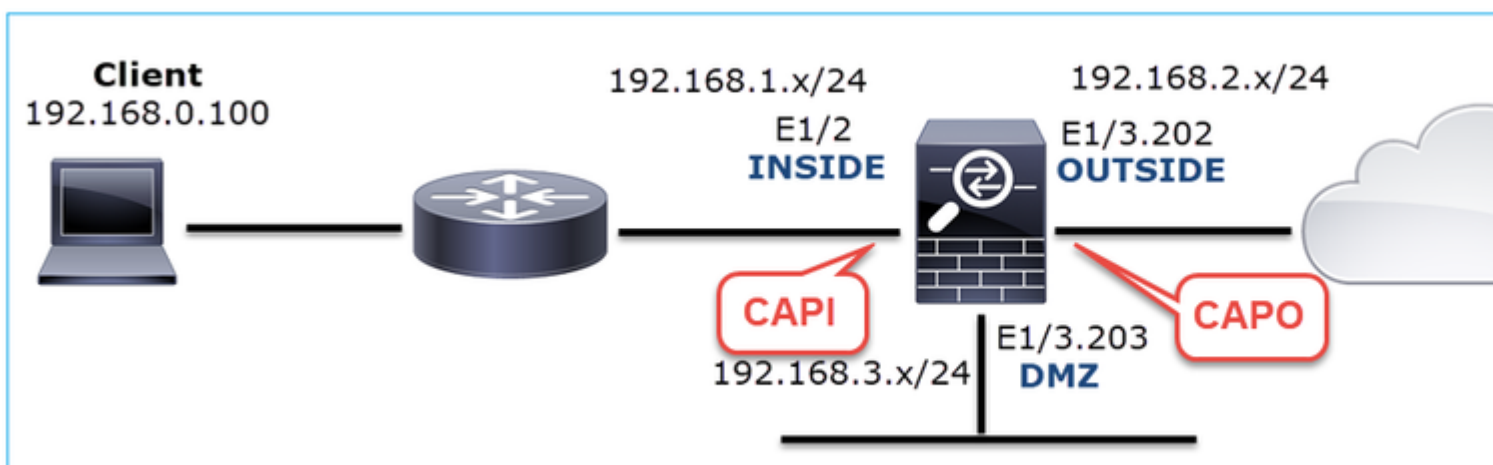
```
<#root>
```

```
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
```



Erfassungen - Nicht-funktionales Szenario:

Dies sind die CAPI-Inhalte.

```
<#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 1
  2: 12:32:23.111307 192.168.0.100.47079 > 10.10.1.100.80: S 2486945841:2486945841(0) win 8192 <mss 1
  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 1
  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 1
  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 1
 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 1
 14: 12:32:32.160956 192.168.0.100.47079 > 10.10.1.100.80: R 3000518858:3000518858(0) win 0
```

```
14 packets shown
```

Dies sind die CAPO-Inhalte:

```
<#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 1386249852:1386249852
  2: 12:32:23.111429 802.1q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: S 3000518857:3000518857
  3: 12:32:23.112405 802.1q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: R 3514091874:3514091874
  4: 12:32:25.858125 802.1q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: S 1386249852:1386249852
  5: 12:32:25.868729 802.1q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: R 2968892337:2968892337
  6: 12:32:26.108240 802.1q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: S 3822259745:3822259745
  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 4294058752:4294058752
  9: 12:32:31.860917 802.1q vlan#202 P0 192.168.0.100.47078 > 10.10.1.100.80: R 1581733941:1581733941
 10: 12:32:32.160102 802.1q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: S 4284301197:4284301197
 11: 12:32:32.160971 802.1q vlan#202 P0 192.168.0.100.47079 > 10.10.1.100.80: R 502906918:502906918(0)
```

```
11 packets shown
```

Die Firewall-Protokolle zeigen Folgendes an:

```
<#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 (192.168.0.100) to OUTSIDE:10.10.1.100/80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_
Oct 13 2019 13:57:36: %FTD-6-302014: Teardown TCP connection 4869 for INSIDE:192.168.0.100/47741 to OUTSIDE:10.10.1.100/80 [RST] Seq=513573017 Win=0 Len=0
```

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 (192.168.0.100) to OUTSIDE:10.10.1.100/80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_
Oct 13 2019 13:57:39: %FTD-6-302014: Teardown TCP connection 4870 for INSIDE:192.168.0.100/47741 to OUTSIDE:10.10.1.100/80 [RST] Seq=513573017 Win=0 Len=0
```

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 (192.168.0.100) to OUTSIDE:10.10.1.100/80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_
Oct 13 2019 13:57:45: %FTD-6-302014: Teardown TCP connection 4871 for INSIDE:192.168.0.100/47741 to OUTSIDE:10.10.1.100/80 [RST] Seq=513573017 Win=0 Len=0
```

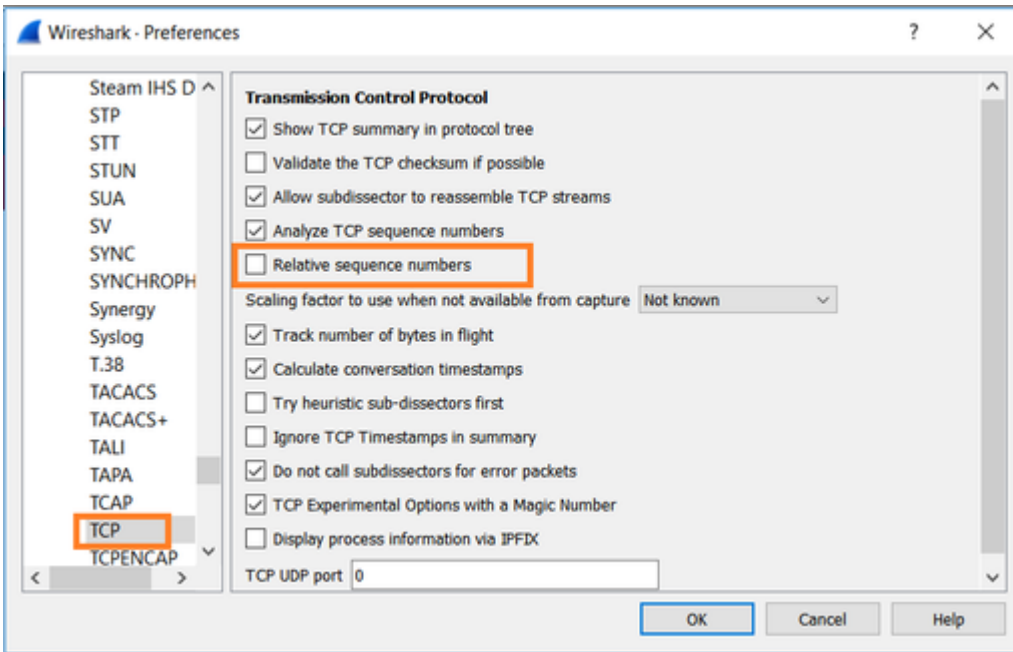
Diese Protokolle zeigen an, dass eine TCP-RST-Nachricht an der Firewall INSIDE-Schnittstelle eingeht.

CAPI-Erfassung in Wireshark:

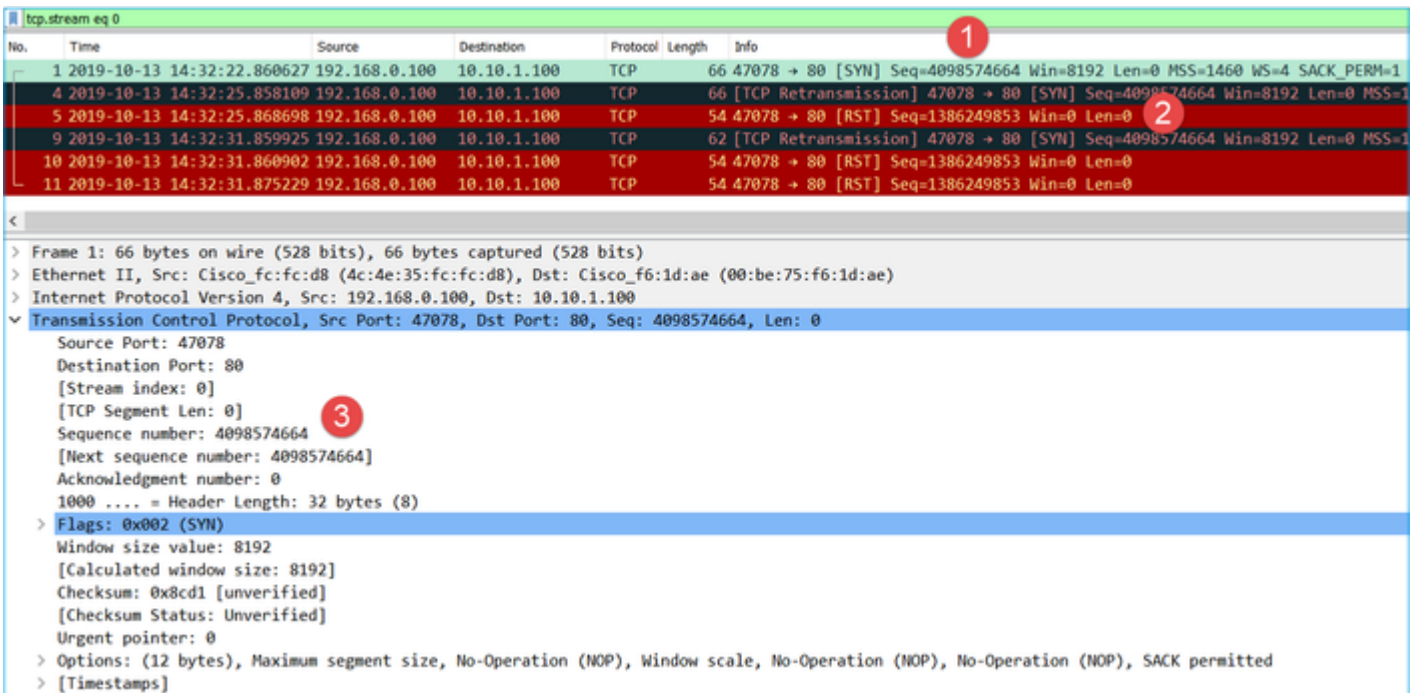
Folgen Sie dem ersten TCP-Stream, wie in der Abbildung dargestellt.

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

Navigieren Sie unter **Wireshark** zu **Edit > Preferences > Protocols > TCP**, und deaktivieren Sie die Option **Relative Sequenznummern** wie im Bild dargestellt.



Dieses Bild zeigt den Inhalt des ersten Flusses bei der CAPI-Erfassung:



Wichtigste Punkte:

1. Der Client sendet ein TCP-SYN-Paket.
2. Der Client sendet ein TCP-RST-Paket.
3. Das TCP-SYN-Paket hat einen Sequenznummernwert gleich 4098574664.

Derselbe Fluss in der CAPO-Erfassung enthält:

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

<

> Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
 > Ethernet II, Src: Cisco_f6:1d:8e (00:be:75:f6:1d:8e), Dst: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8)
 > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 202
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 10.10.1.100
 > Transmission Control Protocol, Src Port: 47078, Dst Port: 80, Seq: 1386249852, Len: 0

Wichtigste Punkte:

1. Der Client sendet ein TCP-SYN-Paket. Die Firewall setzt den ISN nach dem Zufallsprinzip um.
2. Der Client sendet ein TCP-RST-Paket.

Auf der Grundlage der beiden Aufzeichnungen kann der Schluss gezogen werden, dass

- Es gibt keinen TCP-Drei-Wege-Handshake zwischen Client und Server.
- Es gibt eine TCP-RST, die vom Client stammt. Der Wert für die TCP-RST-Sequenznummer bei der CAPI-Erfassung ist 1386249853.

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Erfassen Sie den Client.

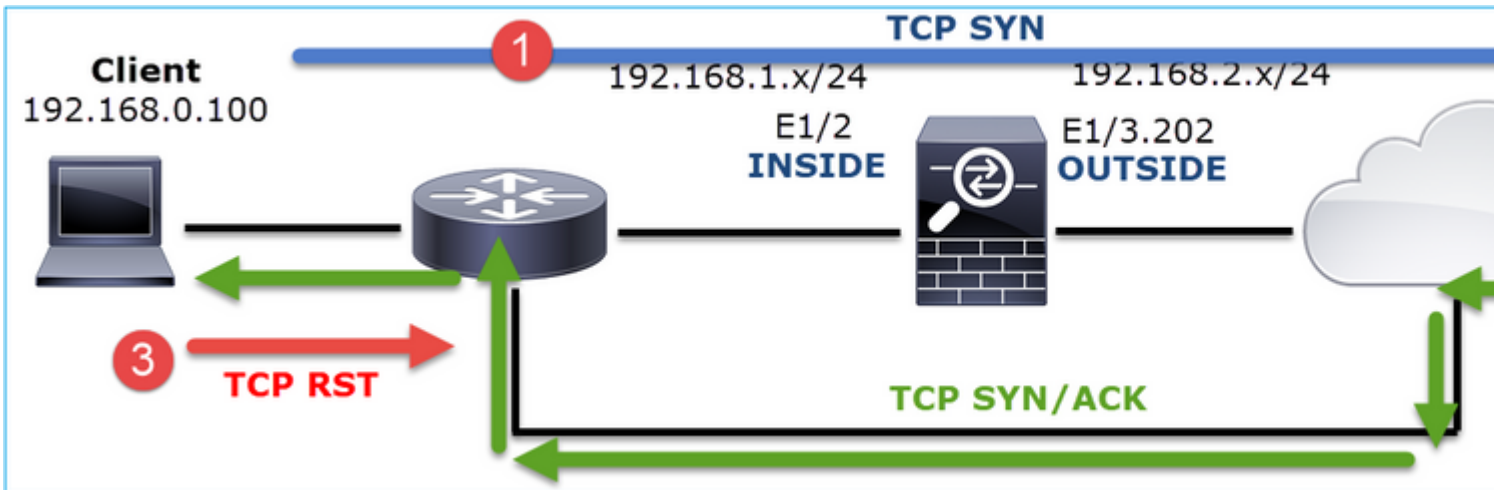
Basierend auf den Aufnahmen, die über die Firewall gesammelt wurden, gibt es deutliche Hinweise auf einen asymmetrischen Datenfluss. Dies basiert darauf, dass der Client eine TCP-RST mit dem Wert 1386249853 (das randomisierte ISN) sendet:

No.	Time	Source	Destination	Protocol	Length	Info
19	6.040337	192.168.0.100	10.10.1.100	TCP	66	47078→80 [SYN] Seq=4098574664
29	9.037499	192.168.0.100	10.10.1.100	TCP	66	[TCP Retransmission] 47078→80 [SYN] Seq=4098574664
30	9.048155	10.10.1.100	192.168.0.100	TCP	66	[TCP ACKed unseen segment] 80→47078 [SYN, A
31	9.048184	192.168.0.100	10.10.1.100	TCP	54	47078→80 [RST] Seq=1386249853 Win=0 Len=0

Wichtigste Punkte:

1. Der Client sendet ein TCP-SYN-Paket. Die Sequenznummer lautet 4098574664 und ist mit der Sequenznummer auf der Firewall INSIDE-Schnittstelle (CAPI) identisch.
2. Es gibt ein TCP SYN/ACK mit der ACK-Nummer 1386249853 (dies wird aufgrund der ISN-Randomisierung erwartet). Dieses Paket wurde bei der Firewall-Erfassung nicht erkannt.
3. Der Client sendet eine TCP-RST, da er eine SYN/ACK mit dem ACK-Zahlenwert 4098574665 erwartet hat, aber den Wert 1386249853 erhalten hat.

Dies kann wie folgt visualisiert werden:

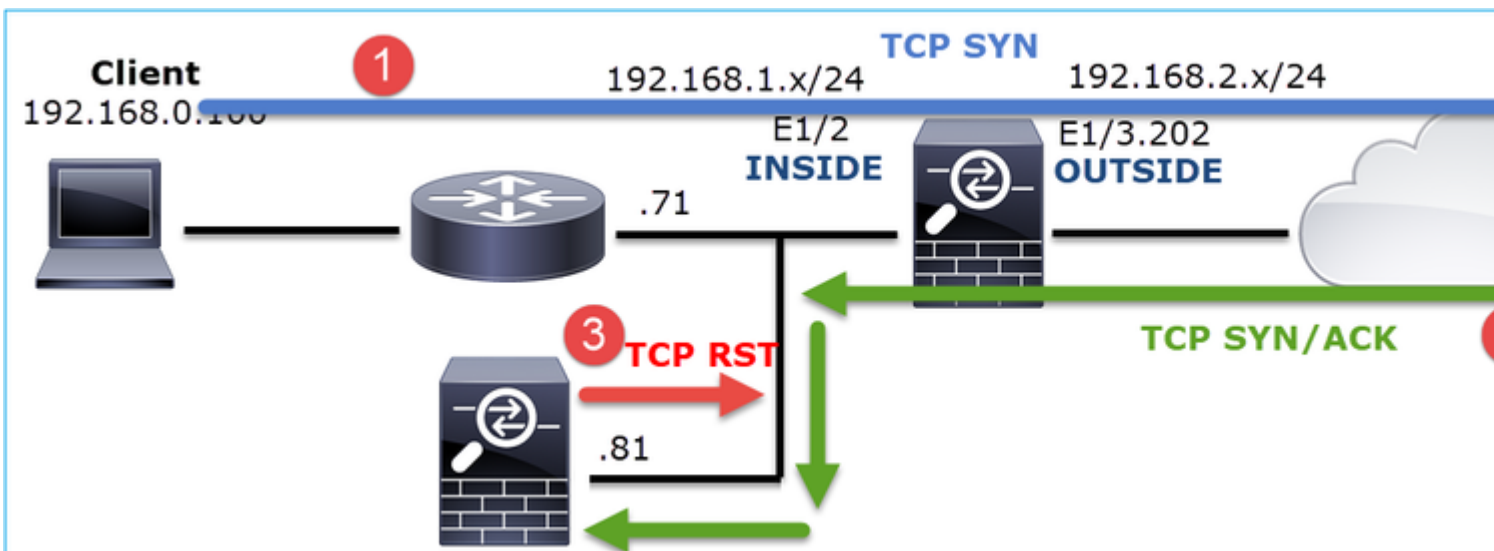


Maßnahme 2: Überprüfen Sie das Routing zwischen Client und Firewall.

Bestätigen Sie Folgendes:

- Die bei den Erfassungen angezeigten MAC-Adressen sind die erwarteten.
- Stellen Sie sicher, dass das Routing zwischen der Firewall und dem Client symmetrisch ist.

Es gibt Szenarien, in denen die RST von einem Gerät stammt, das sich zwischen der Firewall und dem Client befindet, während im internen Netzwerk ein asymmetrisches Routing stattfindet. Ein typischer Fall ist in der Abbildung dargestellt:



In diesem Fall hat die Erfassung diesen Inhalt. Beachten Sie den Unterschied zwischen der Quell-MAC-Adresse des TCP-SYN-Pakets und der Quell-MAC-Adresse der TCP-RST und der Ziel-MAC-Adresse des TCP-SYN/ACK-Pakets:

```
<#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,r
  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,r
  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 8
  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)
```

...

Fall 5: Langsame TCP-Übertragung (Szenario 1)

Problembeschreibung:

Die SFTP-Übertragung zwischen den Hosts 10.11.4.171 und 10.77.19.11 ist langsam. Obwohl die minimale Bandbreite (BW) zwischen den beiden Hosts 100 Mbit/s beträgt, geht die Übertragungsgeschwindigkeit nicht über 5 Mbit/s hinaus.

Gleichzeitig ist die Übertragungsgeschwindigkeit zwischen den Hosts 10.11.2.124 und 172.25.18.134 deutlich höher.

Hintergrundtheorie:

Die maximale Übertragungsgeschwindigkeit für einen einzelnen TCP-Datenstrom wird durch das Bandwidth Delay Product (BDP) bestimmt. Die verwendete Formel wird im Bild angezeigt:

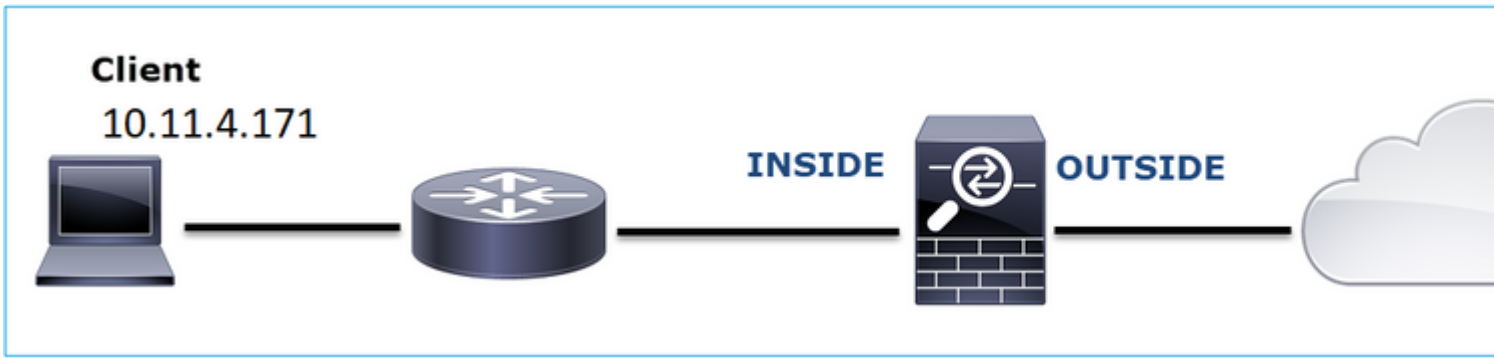
$$\text{Max Single TCP Flow Throughput [bps]} = \frac{\text{TCP Window (Bytes)}}{\text{RTT (Seconds)}} \times 8 \text{ [bits/Byte]}$$

Weitere Informationen zum BDP finden Sie in den Ressourcen unter:

- [Warum nutzt Ihre Anwendung nur 10 Mbit/s Auch wenn die Verbindung 1 Gbit/s beträgt?](#)
- [BRKSEC-3021 - Advanced - Maximierung der Firewall-Leistung](#)

Szenario 1. Langsame Übertragung

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Quelle IP: 10.11.4.171

Ziel-IP: 10.77.19.11

Protokoll: SFTP (FTP über SSH)

Erfassungsanalyse

Erfassung auf FTD LINA-Engine aktivieren:

```
<#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
```

Warnung: LINA-Erfassungen bei FP1xxx- und FP21xx-Erfassungen wirken sich auf die Übertragungsrate des Datenverkehrs aus, der über die FTD übertragen wird. Aktivieren Sie LINA-Aufzeichnungen auf FP1xxx- und FP21xxx-Plattformen nicht, wenn Sie Leistungsprobleme beheben (langsame Übertragung durch FTD). Verwenden Sie stattdessen SPAN oder ein HW-Tap-Gerät, zusätzlich zu den Erfassungen auf dem Quell- und Zielhost. Das Problem ist dokumentiert in Cisco Bug-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.
```

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Berechnung der Round-Trip-Zeit (RTT)

Identifizieren Sie zunächst den Übergabestrom, und folgen Sie ihm:

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

Ändern Sie die Wireshark-Ansicht, um die **Sekunden seit dem zuvor angezeigten Paket** anzuzeigen. Dies erleichtert die Berechnung der RTT:

No.	Time	Source	Destination	Protocol	Length	Window size value	Info
1	0.000000	10.11.4.171	10.77.19.11	TCP	70	49640	39744 → 22 [SYN] Seq=1737026
2	0.072521	10.77.19.11	10.11.4.171	TCP	70	49680	22 → 39744 [SYN, ACK] Seq=83
3	0.000168	10.11.4.171	10.77.19.11	TCP	58	49680	39744 → 22 [ACK] Seq=1737026
4	0.077068	10.77.19.11	10.11.4.171	SSHv2	80	49680	Server: Protocol (SSH-2.0-Su
5	0.000152	10.11.4.171	10.77.19.11	TCP	58	49680	39744 → 22 [ACK] Seq=1737026

Die RTT kann durch Addition der Zeitwerte zwischen 2 Paketvermittlungsstellen (einer zur Quelle und einer zum Ziel) berechnet werden. In diesem Fall zeigt Paket #2 den RTT zwischen der Firewall und dem Gerät an, das das SYN/ACK-Paket (Server) gesendet hat. Paket #3 zeigt die RTT zwischen der Firewall und dem Gerät, das das ACK-Paket gesendet hat (Client). Durch Hinzufügen der beiden Zahlen ergibt sich eine gute Schätzung des End-to-End-RTTs:

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

RTT $\hat{=}$ 80 ms

Berechnung der TCP-Fenstergröße

Erweitern Sie ein TCP-Paket, erweitern Sie den TCP-Header, wählen Sie **Berechnete Fenstergröße** aus, und wählen Sie **Als Spalte übernehmen**:

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

- Source Port: 22
- Destination Port: 39744
- [Stream index: 0]
- [TCP Segment Len: 32]
- Sequence number: 835184024
- [Next sequence number: 835184056]
- Acknowledgment number: 1758069308
- 0101 = Header Length: 20 bytes (5)
- > Flags: 0x018 (PSH, ACK)
- Window size value: 49680
- [Calculated window size: 49680]
- [Window size scaling factor: 1]
- Checksum: 0x2b49 [unverified]
- [Checksum Status: Unverified]
- Urgent pointer: 0

The scaled window size (if scaling has been used) is 49680

Expand Subtrees
Collapse Subtrees
Expand All
Collapse All
Apply as Column

Überprüfen Sie in der Spalte **Berechneter Fenstergrößewert**, um den maximalen Fenstergrößewert während der TCP-Sitzung anzuzeigen. Sie können auch den Spaltennamen auswählen und die Werte sortieren.

Wenn Sie einen Dateidownload testen (**Server > Client**), müssen Sie die vom Server angegebenen Werte überprüfen. Der vom Server angegebene Wert für die maximale Fenstergröße bestimmt die erreichte maximale Übertragungsgeschwindigkeit.

In diesem Fall beträgt die TCP-Fenstergröße $\hat{=}$ 50000 Byte

No.	Time	Source	Destination	Protocol	Length	Calculated window size	Info
24...	0.000091	10.11.4.171	10.77.19.11	TCP	58	49680	39744 → 22 [ACK] Seq=1758069341
24...	0.000077	10.77.19.11	10.11.4.171	TCP	58	49680	22 → 39744 [FIN, ACK] Seq=835184152
24...	0.071605	10.77.19.11	10.11.4.171	TCP	58	49680	22 → 39744 [ACK] Seq=835184152
24...	0.000153	10.11.4.171	10.77.19.11	TCP	58	49680	39744 → 22 [FIN, ACK] Seq=1758069341
24...	0.000443	10.11.4.171	10.77.19.11	SSHv2	90	49680	Client: Encrypted packet (len=32)
24...	0.071666	10.77.19.11	10.11.4.171	SSHv2	154	49680	Server: Encrypted packet (len=96)
24...	0.044050	10.11.4.171	10.77.19.11	TCP	58	49680	39744 → 22 [ACK] Seq=1758069308
24...	0.073605	10.77.19.11	10.11.4.171	SSHv2	90	49680	Server: Encrypted packet (len=32)
24...	0.000747	10.11.4.171	10.77.19.11	SSHv2	90	49680	Client: Encrypted packet (len=32)

Basierend auf diesen Werten und unter Verwendung der Formel Bandwidth Delay Product erhalten Sie die

maximale theoretische Bandbreite, die unter diesen Bedingungen erreicht werden kann: $50000 * 8 / 0,08 = 5$ Mbit/s maximale theoretische Bandbreite.

Dies entspricht dem, was der Client in diesem Fall erlebt.

Überprüfen Sie den Drei-Wege-TCP-Handshake genau. Beide Seiten, und noch wichtiger der Server, kündigen einen Fensterskalierungswert von 0 an, was $2^0 = 1$ bedeutet (keine Fensterskalierung). Dies wirkt sich negativ auf die Übertragungsrate aus:

No.	Time	Source	Destination	Protocol	Length	Window size value	Info
1	0.000000	10.11.4.171	10.77.19.11	TCP	70	49640 39744 → 22	[SYN] Seq=1737026093 Win=49640 L
2	0.072521	10.77.19.11	10.11.4.171	TCP	70	49680 22 → 39744	[SYN, ACK] Seq=835172681 Ack=173


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

An dieser Stelle muss eine Erfassung auf dem Server durchgeführt werden, es muss überprüft werden, ob es derjenige ist, der Fensterskala = 0 ankündigt, und diese neu konfiguriert werden (weitere Informationen hierzu finden Sie in der Serverdokumentation).

Szenario 2. Schnelle Übertragung

Betrachten wir nun das gute Szenario (schnelle Übertragung über dasselbe Netzwerk):

Topologie:



Der Fluss des Interesses:

Quelle IP: 10.11.2.124

Ziel: 172.25.18.134

Protokoll: SFTP (FTP über SSH)

Erfassung auf FTD LINA-Engine aktivieren

<#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
```

Berechnung der Round Trip Time (RTT): In diesem Fall beträgt die RTT $\hat{=}$ 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

Berechnung der TCP-Fenstergröße: Der Server kündigt einen TCP-Fensterskalierungsfaktor von 7 an.

```
> 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]
```

Die TCP-Fenstergröße des Servers beträgt $\hat{=}$ 1600000 Byte:

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

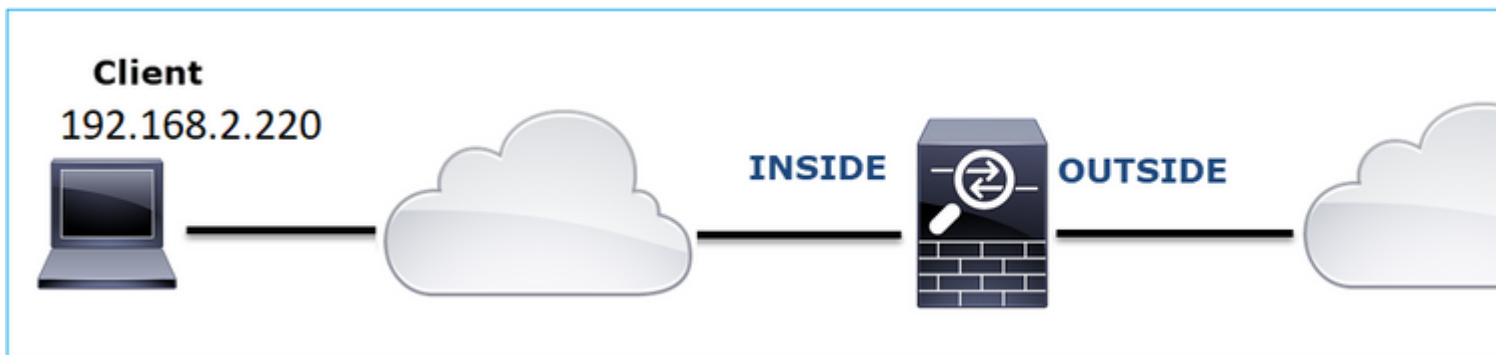
Basierend auf diesen Werten ergibt die Formel für die Bandbreitenverzögerung:

$1600000 \times 8/0,3 = 43 \text{ Mbit/s}$ theoretische maximale Übertragungsgeschwindigkeit

Fall 6: Langsame TCP-Übertragung (Szenario 2)

Problembeschreibung: Die FTP-Dateiübertragung (Download) durch die Firewall ist langsam.

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Quelle IP: 192.168.2.220

Ziel: 192.168.1.220

Protokoll: FTP

Erfassungsanalyse

Aktivieren Sie Aufnahmen auf der FTD LINA-Engine.

```
<#root>
```

```
firepower#
```

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

```
firepower#
```

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

Wählen Sie ein FTP-DATA-Paket aus, und folgen Sie dem FTP Data Channel auf FTD INSIDE Capture (CAPI):

75	0.000412	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670018383
76	0.000518	192.168.1.220	192.168.2.220	FTP-DATA	(PASV) (RETR file15mb)
77	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	Mark/Unmark Packet (PASV) (RETR file15mb)
78	0.000046	192.168.1.220	192.168.2.220	FTP-DATA	Ignore/Unignore Packet not captured] FTP Data: 124
79	0.000015	192.168.1.220	192.168.2.220	FTP-DATA	Set/Unset Time Reference (PASV) (RETR file15mb)
80	0.000107	192.168.2.220	192.168.1.220	TCP	Time Shift... q=1884231612 Ack=2670019631
81	0.000092	192.168.2.220	192.168.1.220	TCP	Packet Comment... q=1884231612 Ack=2670020879
82	0.000091	192.168.2.220	192.168.1.220	TCP	4494 → 2388 [ACK] Seq=188423
83	0.000015	192.168.2.220	192.168.1.220	TCP	4494 → 2388 [ACK] Seq=188423
84	0.000321	192.168.1.220	192.168.2.220	FTP-DATA	Apply as Filter (PASV) (RETR file15mb)
85	0.000061	192.168.1.220	192.168.2.220	FTP-DATA	Prepare a Filter (PASV) (RETR file15mb)
86	0.000153	192.168.2.220	192.168.1.220	TCP	Conversation Filter 4494 → 2388 [ACK] Seq=188423
87	0.000122	192.168.2.220	192.168.1.220	TCP	Colorize Conversation 4494 → 2388 [ACK] Seq=188423
88	0.918415	192.168.1.220	192.168.2.220	TCP	SCTP 38 → 54494 [ACK] Seq=2670020
89	0.000397	192.168.2.220	192.168.1.220	TCP	Follow TCP Stream =2670027119
90	0.000869	192.168.1.220	192.168.2.220	FTP-DATA	FTP Stream e15mb)

Inhalt des FTP-DATA-Streams:

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 TSres=0
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
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
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=4
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=4
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=3577291510 TSecr=4
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=3577291510 TSecr=4
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=3577291510 TSecr=4
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=4
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=4
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=4
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=3577291821 TSecr=4
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=3577291821 TSecr=4
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=4
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=4
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=4
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=4
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=3577292742 TSecr=4
60	0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

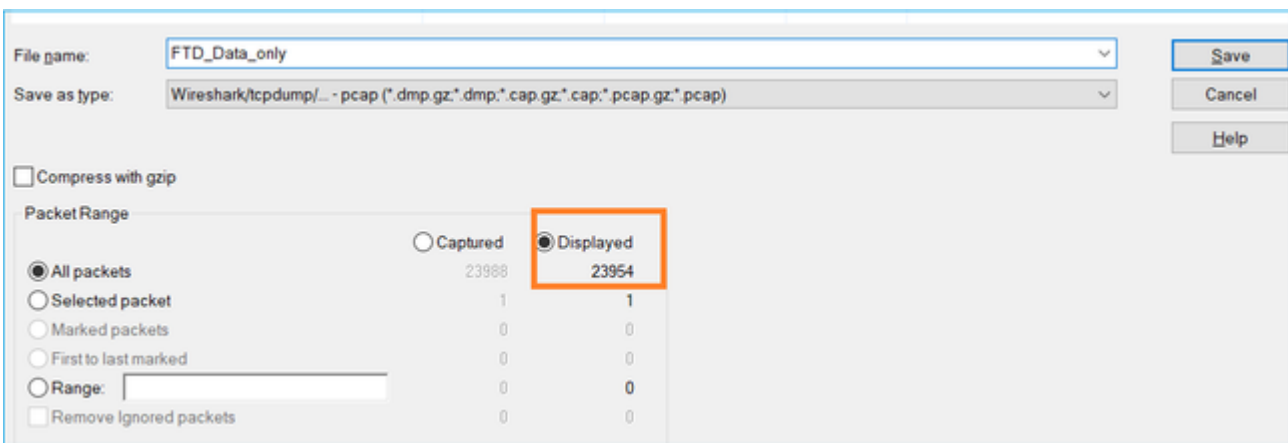
Der CAPO-Inhalt:

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 T
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
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 HS=256 SACK
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=4
38	0.001328	192.168.1.220	192.168.2.220	FTP-DATA	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)
40	0.000641	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSe
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 TSva
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 TSva
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 TS
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=4
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=4
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 TSva
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 TSva
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 TS
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=4
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=4
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=4
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 TSva
64	0.000107	192.168.1.220	192.168.2.220	FTP-DATA	1314	FTP Data: 1248 bytes (PASV) (RETR file15mb)

Wichtigste Punkte:

1. Es gibt TCP Out-Of-Order (OOO)-Pakete.
2. Es findet eine TCP-Neuübertragung statt.
3. Es gibt einen Hinweis auf einen Paketverlust (verlorene Pakete).

Tipp: Speichern Sie die Aufzeichnungen, während Sie zu **Datei > Angegebene Pakete exportieren** navigieren. Speichern Sie dann nur den **angezeigten** Paketbereich.



Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Identifizieren Sie den Ort für den Paketverlust.

In Fällen wie diesem müssen Sie gleichzeitig aufzeichnen und die divide-and-conquer-Methode verwenden, um die Netzwerksegmente zu identifizieren, die einen Paketverlust verursachen. Aus Sicht der Firewall gibt es drei Hauptszenarien:

1. Der Paketverlust wird durch die Firewall selbst verursacht.
2. Der Paketverlust wird Downstream zum Firewall-Gerät verursacht (Richtung vom Server zum Client).
3. Der Paketverlust wird Upstream zum Firewall-Gerät verursacht (Richtung vom Client zum Server).

Paketverlust durch die Firewall: Um festzustellen, ob der Paketverlust durch die Firewall verursacht wird, muss die Eingangserfassung mit der Ausgangserfassung verglichen werden. Es gibt viele Möglichkeiten, zwei verschiedene Aufnahmen zu vergleichen. Dieser Abschnitt zeigt eine Möglichkeit, diese Aufgabe durchzuführen.

Verfahren zum Vergleichen von 2 Erfassungen zur Identifizierung des Paketverlusts

Schritt 1: Stellen Sie sicher, dass die beiden Erfassungen Pakete aus demselben Zeitfenster enthalten. Das bedeutet, dass bei einer Erfassung keine Pakete erfasst werden dürfen, die vor oder nach der anderen Erfassung erfasst wurden. Es gibt mehrere Möglichkeiten, dies zu tun:

- Überprüfen Sie den ersten und letzten Wert der Paket-IP-Identifikation (ID).
- Überprüfen Sie den ersten und letzten Paketzeitstempelwert.

In diesem Beispiel können Sie sehen, dass die ersten Pakete jeder Erfassung die gleichen IP-ID-Werte haben:

No.	Time	Source	Destination	Protocol	Length	Identification	Info
1	2019-10-16 16:13:44.169394	192.168.2.220	192.168.1.220	TCP	74	0x0a34 (2612)	54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TS...
2	2019-10-16 16:13:45.195958	192.168.2.220	192.168.1.220	TCP	74	0x0a35 (2613)	[TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=...
3	2019-10-16 16:13:47.177542	192.168.1.220	192.168.2.220	TCP	74	0x151f (5407)	2388 → 54494 [SYN, ACK] Seq=2669989678 Ack=1884231612 Win=8192 Len=0 MSS=...
4	2019-10-16 16:13:47.178030	192.168.2.220	192.168.1.220	TCP	66	0x0a36 (2614)	
5	2019-10-16 16:13:47.179647	192.168.1.220	192.168.2.220	TCP	1314	0x1521 (5409)	
6	2019-10-16 16:13:47.179998	192.168.2.220	192.168.1.220	TCP	66	0x0a37 (2615)	
7	2019-10-16 16:13:47.180456	192.168.1.220	192.168.2.220	TCP	1314	0x1523 (5411)	
8	2019-10-16 16:13:47.180517	192.168.1.220	192.168.2.220	TCP	1314	0x1524 (5412)	
9	2019-10-16 16:13:47.180715	192.168.2.220	192.168.1.220	TCP	78	0x0a38 (2616)	
10	2019-10-16 16:13:47.180792	192.168.2.220	192.168.1.220	TCP	78	0x0a39 (2617)	
11	2019-10-16 16:13:47.489888	192.168.1.220	192.168.2.220	TCP	1314	0x1525 (5413)	
12	2019-10-16 16:13:47.490376	192.168.2.220	192.168.1.220	TCP	66	0x0a3a (2618)	
13	2019-10-16 16:13:47.490865	192.168.1.220	192.168.2.220	TCP	1314	0x1526 (5414)	
14	2019-10-16 16:13:47.490910	192.168.1.220	192.168.2.220	TCP	1314	0x1528 (5416)	
15	2019-10-16 16:13:47.490987	192.168.1.220	192.168.2.220	TCP	1314	0x1529 (5417)	
16	2019-10-16 16:13:47.491231	192.168.2.220	192.168.1.220	TCP	66	0x0a3b (2619)	
17	2019-10-16 16:13:47.491261	192.168.2.220	192.168.1.220	TCP	78	0x0a3c (2620)	
18	2019-10-16 16:13:47.491765	192.168.1.220	192.168.2.220	TCP	1314	0x152a (5418)	
19	2019-10-16 16:13:47.492024	192.168.2.220	192.168.1.220	TCP	78	0x0a3d (2621)	
20	2019-10-16 16:13:48.410150	192.168.1.220	192.168.2.220	TCP	1314	0x152e (5422)	
21	2019-10-16 16:13:48.411050	192.168.2.220	192.168.1.220	TCP	66	0x0a3e (2622)	
22	2019-10-16 16:13:48.411569	192.168.1.220	192.168.2.220	TCP	1314	0x152f (5423)	
23	2019-10-16 16:13:48.411630	192.168.1.220	192.168.2.220	TCP	1314	0x1530 (5424)	
24	2019-10-16 16:13:48.411645	192.168.1.220	192.168.2.220	TCP	1314	0x1532 (5426)	
25	2019-10-16 16:13:48.411660	192.168.1.220	192.168.2.220	TCP	1314	0x1533 (5427)	
26	2019-10-16 16:13:48.411859	192.168.2.220	192.168.1.220	TCP	66	0x0a3f (2623)	
27	2019-10-16 16:13:48.412088	192.168.2.220	192.168.1.220	TCP	66	0x0a40 (2624)	

Falls sie nicht identisch sind:

1. Vergleichen Sie die Zeitstempel des ersten Pakets jeder Erfassung.
2. Von der Erfassung mit dem neuesten Zeitstempel einen Filter abrufen und den Zeitstempelfilter von == auf >= (das erste Paket) und <= (das letzte Paket) ändern, z.B.:

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

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

Encapsulation type: Ethernet (1)

Arrival Time: Oct 16, 2019 16:13:43.245638000 seconds

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1571235223.245638000 seconds

[Time delta from previous captured frame: 0.000000000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 0.000000000 seconds]

Frame Number: 2

Frame Length: 74 bytes (592 bits)

Capture Length: 74 bytes (592 bits)

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

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

3. Exportieren Sie die angegebenen Pakete in eine neue Erfassung, wählen Sie **Datei > Angegebene Pakete exportieren**, und speichern Sie dann die **angezeigten** Pakete. An diesem Punkt müssen beide Erfassungen Pakete enthalten, die dasselbe Zeitfenster abdecken. Sie können nun den Vergleich der 2 Aufnahmen starten.

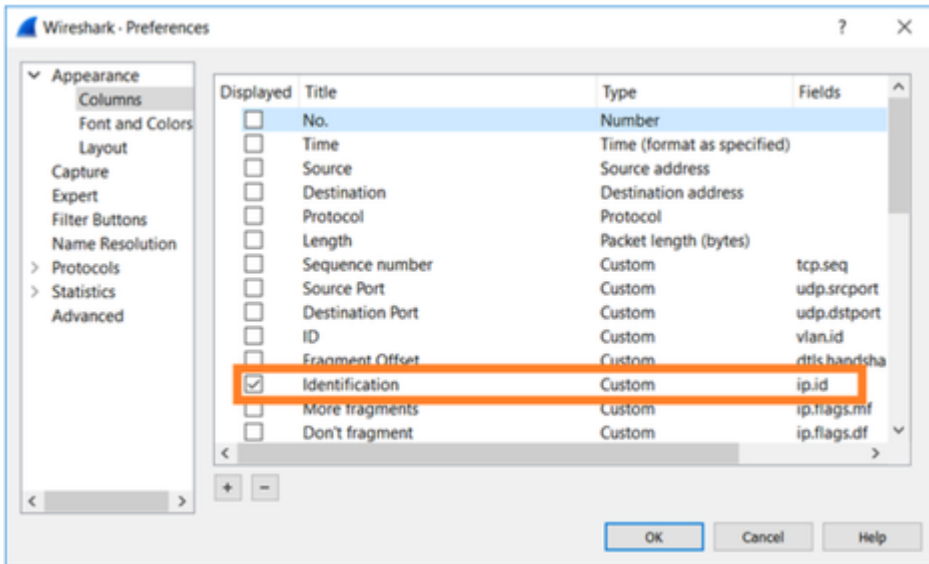
Schritt 2: Geben Sie an, welches Paketfeld für den Vergleich zwischen den beiden Erfassungen verwendet wird. Beispiel für verwendbare Felder:

- IP-Identifizierung
- RTP-Sequenznummer
- ICMP-Sequenznummer

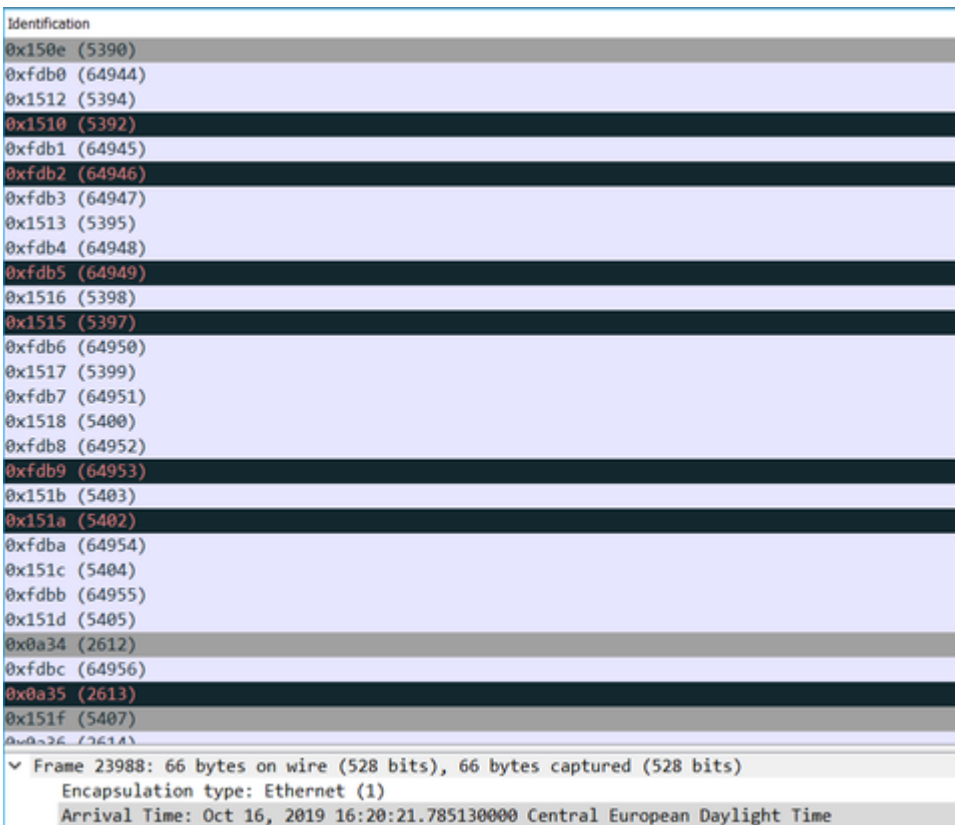
Erstellen Sie eine Textversion jeder Erfassung, die das Feld für jedes Paket enthält, das Sie in Schritt 1 angegeben haben. Um dies zu tun, lassen Sie nur die Spalte von Interesse, zum Beispiel, wenn Sie Pakete basierend auf IP-Identifizierung vergleichen wollen dann ändern Sie die Erfassung wie im Bild dargestellt.

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

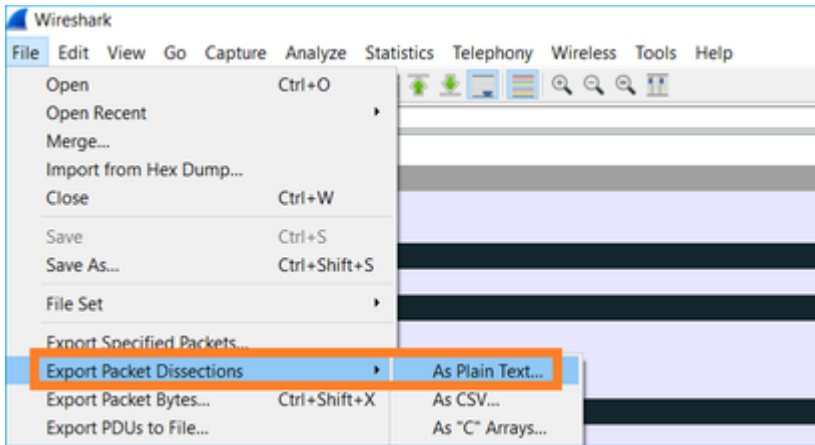
No.	Time	Source	Destination	Protocol	Length	Info
2	2019-10-16 16:13:43.245638	192.168.1.220	192.168.2.220	TCP	74	21 → 38400
3	2019-10-16 16:13:43.245867	192.168.2.220	192.168.1.220	TCP	66	38400 → 21
4	2019-10-16 16:13:43.558259	192.168.1.220	192.168.2.220	FTP	229	Response
5	2019-10-16 16:13:43.558274	192.168.1.220	192.168.2.220	TCP	126	[TCP Out-



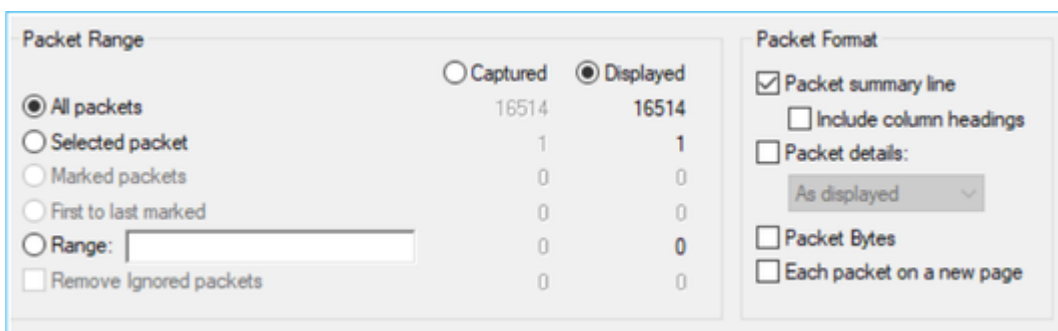
Ergebnis:



Schritt 3: Erstellen Sie eine Textversion der Erfassung (**Datei > Paketdissektionen exportieren > Als Klartext...**), wie im Bild gezeigt:



Deaktivieren Sie die Optionen **Spaltenüberschriften** und **Paketdetails einschließen**, um nur die Werte des angezeigten Felds zu exportieren, wie in der Abbildung dargestellt:



Schritt 4: Sortieren Sie die Pakete in den Dateien. Dazu können Sie den Linux-Befehl **sort** verwenden:

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

Schritt 5: Verwenden Sie ein Textvergleichstool (z. B. WinMerge) oder den Linux-Befehl **diff**, um die Unterschiede zwischen den beiden Aufnahmen zu ermitteln.

0x0a3d (2621)	0x0a3d (2621)
0x0a3e (2622)	0x0a3e (2622)
0x0a3f (2623)	0x0a3f (2623)
0x0a40 (2624)	0x0a40 (2624)
0x0a41 (2625)	0x0a41 (2625)
0x0a42 (2626)	0x0a42 (2626)
0x0a43 (2627)	0x0a43 (2627)
0x0a44 (2628)	0x0a44 (2628)
0x0a45 (2629)	0x0a45 (2629)
0x0a46 (2630)	0x0a46 (2630)
0x0a47 (2631)	0x0a47 (2631)
0x0a48 (2632)	0x0a48 (2632)
0x0a49 (2633)	0x0a49 (2633)
0x0a4a (2634)	0x0a4a (2634)
0x0a4b (2635)	0x0a4b (2635)
0x0a4c (2636)	0x0a4c (2636)
0x0a4d (2637)	0x0a4d (2637)
0x0a4e (2638)	0x0a4e (2638)
0x0a4f (2639)	0x0a4f (2639)

WinMerge

The selected files are identical.

Don't display this message again.

Ok

Ln: 27 Col: 14/14 Ch: 14/14 1252 Win Ln: 23955 Col: 1/1 Ch: 1/1

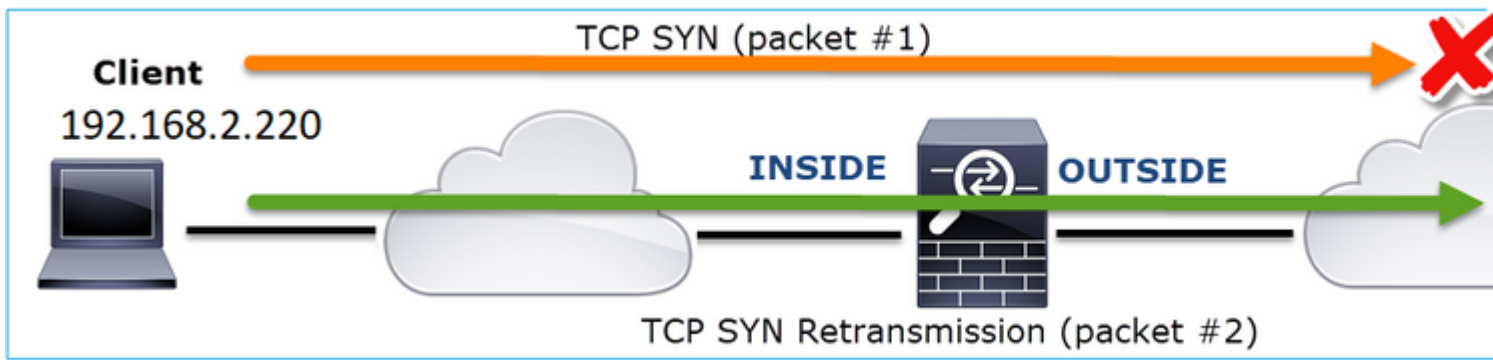
In diesem Fall sind die CAPI- und CAPO-Erfassung für den FTP-Datenverkehr identisch. Dies beweist, dass der Paketverlust nicht durch die Firewall verursacht wurde.

Identifizieren von Upstream-/Downstream-Paketverlusten

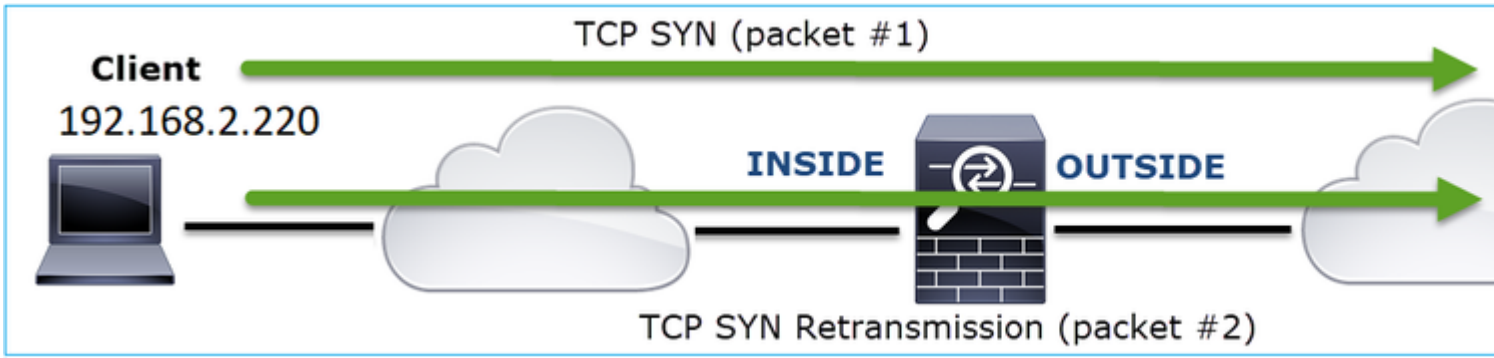
No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-16 16:13:44.169516	192.168.2.220	192.168.1.220	TCP	74	54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0
2	2019-10-16 16:13:45.196050	192.168.2.220	192.168.1.220	TCP	74	[TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681
3	2019-10-16 16:13:47.177450	192.168.1.220	192.168.2.220	TCP	74	2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682
4	2019-10-16 16:13:47.178060	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912
5	2019-10-16 16:13:47.179388	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224316912 Ack=2157030682
6	2019-10-16 16:13:47.180029	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160
7	2019-10-16 16:13:47.180410	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224318160
8	2019-10-16 16:13:47.180456	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224320656 Ack=2157030682
9	2019-10-16 16:13:47.180746	192.168.2.220	192.168.1.220	TCP	78	[TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682
10	2019-10-16 16:13:47.180822	192.168.2.220	192.168.1.220	TCP	78	[TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682
11	2019-10-16 16:13:47.489827	192.168.1.220	192.168.2.220	TCP	1314	[TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224318160
12	2019-10-16 16:13:47.490407	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904
13	2019-10-16 16:13:47.490819	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224321904 Ack=2157030682
14	2019-10-16 16:13:47.490880	192.168.1.220	192.168.2.220	TCP	1314	[TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224321904
15	2019-10-16 16:13:47.490956	192.168.1.220	192.168.2.220	TCP	1314	2388 → 54494 [ACK] Seq=2224325648 Ack=2157030682
16	2019-10-16 16:13:47.491246	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152

Wichtigste Punkte:

1. Dieses Paket ist eine TCP-Neuübertragung. Dabei handelt es sich insbesondere um ein TCP-SYN-Paket, das vom Client für passive FTP-Daten an den Server gesendet wird. Da der Client das Paket erneut sendet und Sie die anfängliche SYN (Paket #1) sehen können, ging das Paket Upstream zur Firewall verloren.



In diesem Fall besteht die Möglichkeit, dass das SYN-Paket den Server erreicht hat, das SYN/ACK-Paket jedoch auf dem Rückweg verloren ging:



2. Es liegt ein Paket vom Server vor, und Wireshark hat festgestellt, dass das vorherige Segment nicht gesehen/erfasst wurde. Da das nicht erfasste Paket vom Server an den Client gesendet wurde und nicht in der Firewall-Erfassung zu sehen war, ging das Paket zwischen dem Server und der Firewall verloren.



Dies weist auf einen Paketverlust zwischen dem FTP-Server und der Firewall hin.

Maßnahme 2: Nehmen Sie zusätzliche Aufzeichnungen vor.

Nehmen Sie zusätzliche Aufnahmen zusammen mit Aufnahmen an den Endpunkten. Versuchen Sie, die divide-and-conquer-Methode anzuwenden, um das problematische Segment, das den Paketverlust verursacht, weiter zu isolieren.

No.	Time	Source	Destination	Protocol	Length	Info
155	2019-10-16 16:13:51.749845	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
156	2019-10-16 16:13:51.749860	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
157	2019-10-16 16:13:51.749872	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
158	2019-10-16 16:13:51.750722	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157
159	2019-10-16 16:13:51.750744	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
160	2019-10-16 16:13:51.750768	192.168.2.220	192.168.1.220	TCP	66	54494 → 2388 [ACK] Seq=2157
161	2019-10-16 16:13:51.750782	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
162	2019-10-16 16:13:51.751001	192.168.2.220	192.168.1.220	TCP	78	[TCP Dup ACK 160#1] 54494 →
163	2019-10-16 16:13:51.751024	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
164	2019-10-16 16:13:51.751378	192.168.2.220	192.168.1.220	TCP	78	[TCP Dup ACK 160#2] 54494 →
165	2019-10-16 16:13:51.751402	192.168.1.220	192.168.2.220	FTP-DA...	1314	FTP Data: 1248 bytes (PASV)
166	2019-10-16 16:13:51.751622	192.168.2.220	192.168.1.220	TCP	78	[TCP Dup ACK 160#3] 54494 →
167	2019-10-16 16:13:51.751648	192.168.1.220	192.168.2.220	FTP-DA...	1314	[TCP Fast Retransmission]


```

> Frame 167: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits) on interface 0
> Ethernet II, Src: Vmware_30:2b:78 (00:0c:29:30:2b:78), Dst: Cisco_9d:89:9b (50:3d:e5:9d:89:9b)
> Internet Protocol Version 4, Src: 192.168.1.220, Dst: 192.168.2.220
> Transmission Control Protocol, Src Port: 2388, Dst Port: 494, Seq: 2224386800, Ack: 2157030682, Len: 1248
  FTP Data (1248 bytes data)
  [Setup frame: 33]
  [Setup method: PASV]
  [Command: RETR file15mb]
  Command frame: 40
  [Current working directory: /]
> Line-based text data (1 lines)

```

Wichtigste Punkte:

1. Der Empfänger (in diesem Fall der FTP-Client) verfolgt die eingehenden TCP-Sequenznummern. Wenn ein Paket nicht erkannt wurde (eine erwartete Sequenznummer wurde übersprungen), wird ein ACK-Paket mit der ACK='erwarteten Sequenznummer, die übersprungen wurde' generiert. In diesem Beispiel ist Ack=2224386800.
2. Die Dup ACK löst eine schnelle TCP-Neuübertragung aus (Neuübertragung innerhalb von 20 ms nach Empfang einer Duplikat-ACK).

Was bedeuten doppelte ACKs?

- Ein paar doppelte ACKs, aber keine tatsächlichen Neuübertragungen deuten darauf hin, dass es wahrscheinlicher ist, dass Pakete nicht in der richtigen Reihenfolge ankommen.
- Doppelte ACKs und anschließende erneute Übertragungen weisen auf einen gewissen Paketverlust hin.

Maßnahme 3: Berechnen Sie die Verarbeitungszeit der Firewall für übertragene Pakete.

Wenden Sie die gleiche Erfassung auf zwei verschiedene Schnittstellen an:

```

<#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

```

Exportieren der Erfassungsüberprüfung der Zeitdifferenz zwischen Eingangs- und Ausgangspaketen

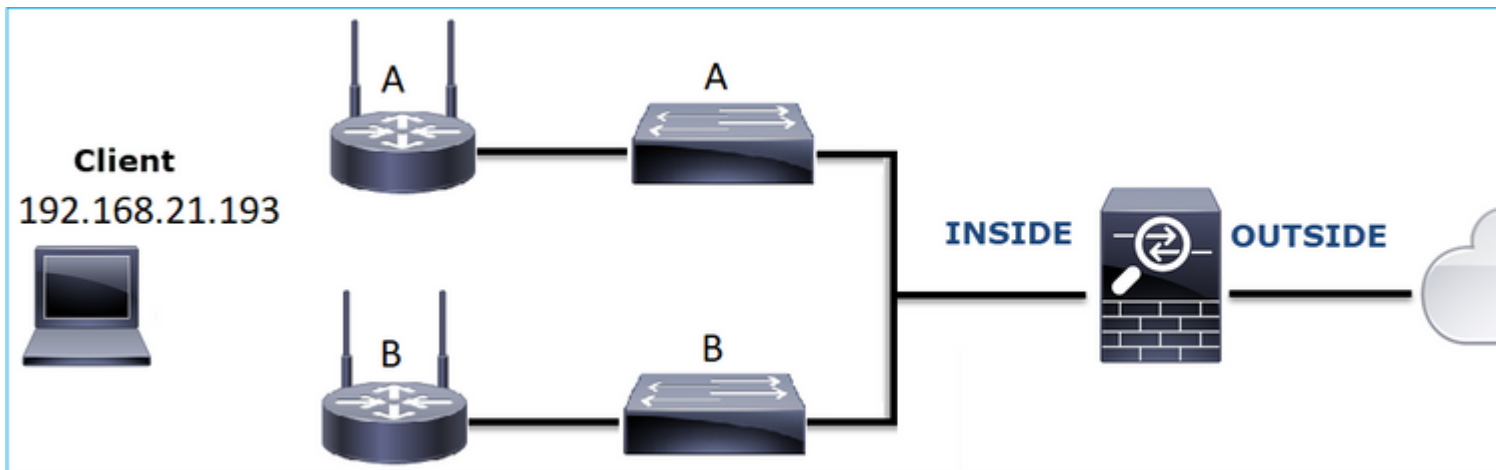
Fall 7: TCP-Verbindungsproblem (Paketbeschädigung)

Problembeschreibung:

Der Wireless-Client (192.168.21.193) versucht, eine Verbindung zu einem Zielsever herzustellen (192.168.14.250 - HTTP). Es gibt zwei verschiedene Szenarien:

- Wenn der Client eine Verbindung mit Access Point (AP) 'A' herstellt, funktioniert die HTTP-Verbindung nicht.
- Wenn der Client eine Verbindung mit Access Point (AP) 'B' herstellt, funktioniert die HTTP-Verbindung.

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Quelle IP: 192.168.21.193

Ziel: 192.168.14.250

Protokoll: TCP 80

Erfassungsanalyse

Erfassung auf FTD LINA-Engine aktivieren:

```
<#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

Erfassungen - Funktionsszenario:

Als Basislinie ist es immer sehr nützlich, Aufzeichnungen aus einem zweifelsfrei funktionierenden Szenario zu haben.

Dieses Bild zeigt die Erfassung der NGFW INSIDE-Schnittstelle.

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=
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
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=6
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=6
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=6
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=6
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

Dieses Bild zeigt die Erfassung der NGFW-OUTSIDE-Schnittstelle.

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=
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
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=6
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=6
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=6
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=6
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

Wichtigste Punkte:

1. Die beiden Aufnahmen sind fast identisch (man denke an die ISN-Randomisierung).
2. Es gibt keine Anzeichen für einen Paketverlust.
3. Keine Out-of-Order (OOB)-Pakete
4. Es gibt 3 HTTP GET-Anforderungen. Die erste erhält die Nummer 404 "Nicht gefunden", die zweite die Nummer 200 "OK" und die dritte die Nummer 304 "Nicht geändert".

Aufnahmen - Szenario mit bekannter Störung:

Der CAPI-Inhalt (Ingress Capture).

No.	Time	Source	Destination	Protocol	Length	Info
1	2013-08-08 15:33:31.909193	192.168.21.193	192.168.14.250	TCP	66	3072 → 80 [SYN] Seq=4231766828 Win=65535 Len=0 MSS=
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
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=6
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=4231
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=867
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=
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=6
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=423
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=867
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=
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
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=
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=213083
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=
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=299
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]
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=299

Wichtigste Punkte:

1. Es gibt einen Drei-Wege-TCP-Handshake.
2. Es gibt TCP-Neuübertragungen und Hinweise auf einen Paketverlust.
3. Es gibt ein Paket (TCP ACK), das von Wireshark als **fehlerhaft** identifiziert wird.

Dieses Bild zeigt den CAPO-Inhalt (Egress Capture).

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=
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
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=6
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=230
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=26
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] Se
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=6
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=230
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=26
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 MS
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=273121942
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=
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=27312
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=
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=24
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]
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=24

Wichtigste Punkte:

Die beiden Aufnahmen sind fast identisch (man denke an die ISN-Randomisierung):

1. Es gibt einen Drei-Wege-TCP-Handshake.
2. Es gibt TCP-Neuübertragungen und Hinweise auf einen Paketverlust.
3. Es gibt ein Paket (TCP ACK), das von Wireshark als **fehlerhaft** identifiziert wird.

Überprüfen Sie das fehlerhafte Paket:

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=
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=4231
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


```

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

```


0000	58 8d 09 61 cc 9b ec 1a 59 63 90 f3 81 00 00 14	X..a....Yc.....
0010	08 00 45 00 00 2a 7f 1d 40 00 80 06 d5 a4 c0 a8	..E..*..@.....
0020	15 c1 c0 a8 0e fa 0c 00 00 50 fc 3b a7 0d 33 b6P;.-3.
0030	28 98 50 10 ff ff 01 bf 00 00 00 00	(.P.....

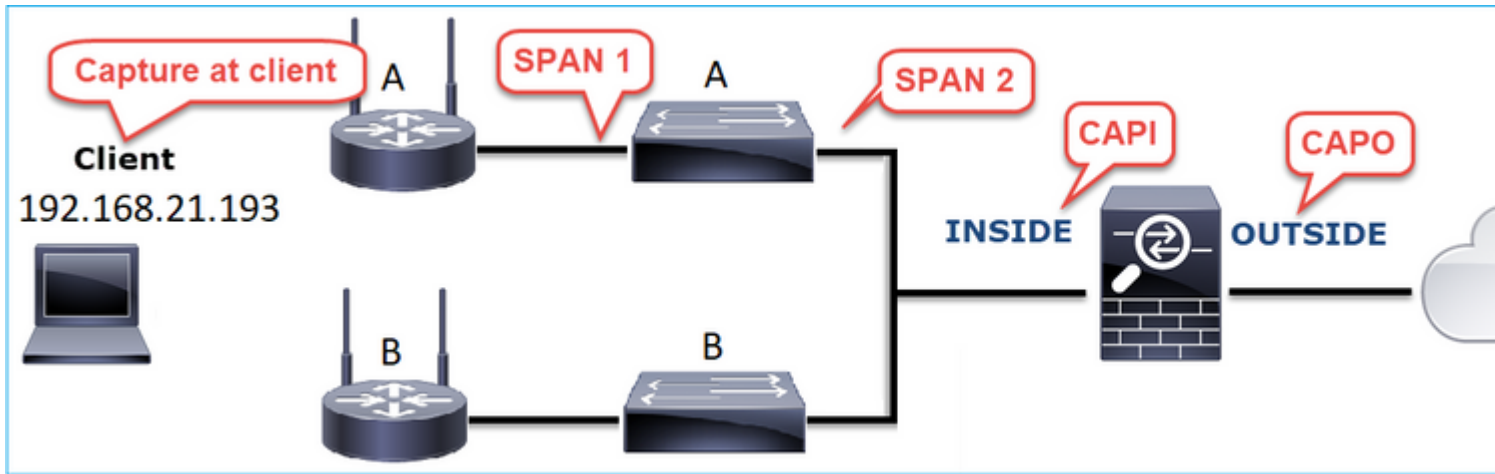
Wichtigste Punkte:

1. Das Paket wird von Wireshark als fehlerhaft identifiziert.
2. Er hat eine Länge von 2 Byte.
3. Es gibt eine TCP-Nutzlast von 2 Byte.
4. Die Nutzlast beträgt 4 zusätzliche Nullen (00 00).

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Machen Sie zusätzliche Aufnahmen. Integrieren Sie Erfassungen an den Endpunkten, und versuchen Sie nach Möglichkeit, die divide-and-conquer-Methode anzuwenden, um die Quelle der Paketbeschädigung zu isolieren. Beispiel:

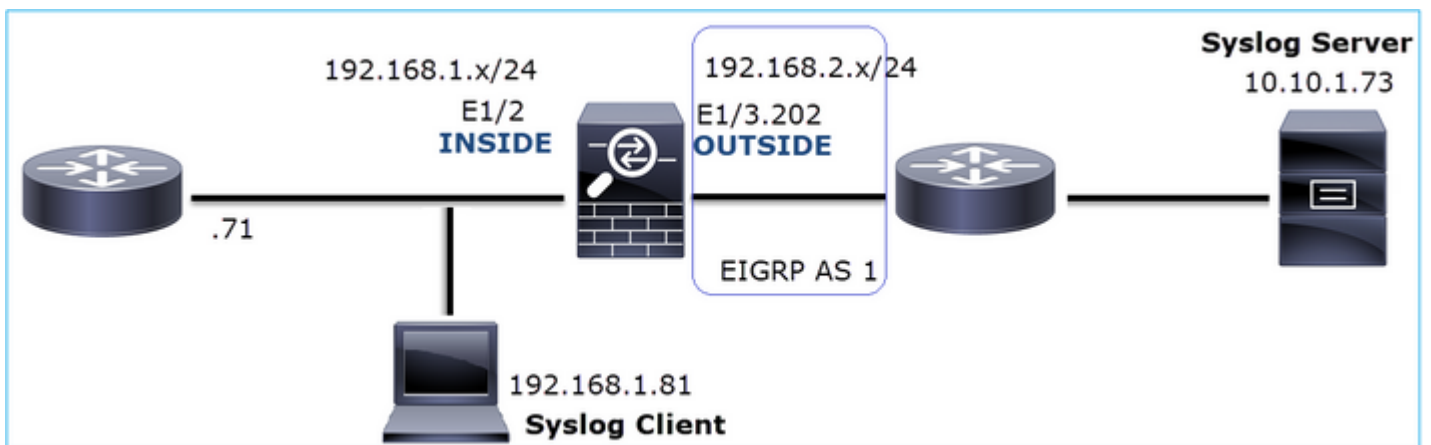


In diesem Fall wurden die 2 zusätzlichen Byte durch den Switch-Schnittstellentreiber "A" hinzugefügt, und die Lösung bestand darin, den Switch zu ersetzen, der die Beschädigung verursacht.

Fall 8: UDP-Verbindungsproblem (fehlende Pakete)

Problembeschreibung: Syslog-Meldungen (UDP 514) werden auf dem Ziel-Syslog-Server nicht erkannt.

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Quelle IP: 192.168.1.81

Ziel-IP: 10.10.1.73

Protokoll: UDP 514

Erfassungsanalyse

Erfassung auf FTD LINA-Engine aktivieren:

<#root>


```
firepower#
```

```
capture CAPI int INSIDE trace match udp host 192.168.1.81 host 10.10.1.73 eq 514
```

```
firepower#
```

```
capture CAPO int OUTSIDE match udp host 192.168.1.81 host 10.10.1.73 eq 514
```

FTD-Erfassungen zeigen keine Pakete an:

```
<#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
```

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Überprüfen Sie die FTD-Verbindungstabelle.

Um eine bestimmte Verbindung zu überprüfen, können Sie folgende Syntax verwenden:

```
<#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
```

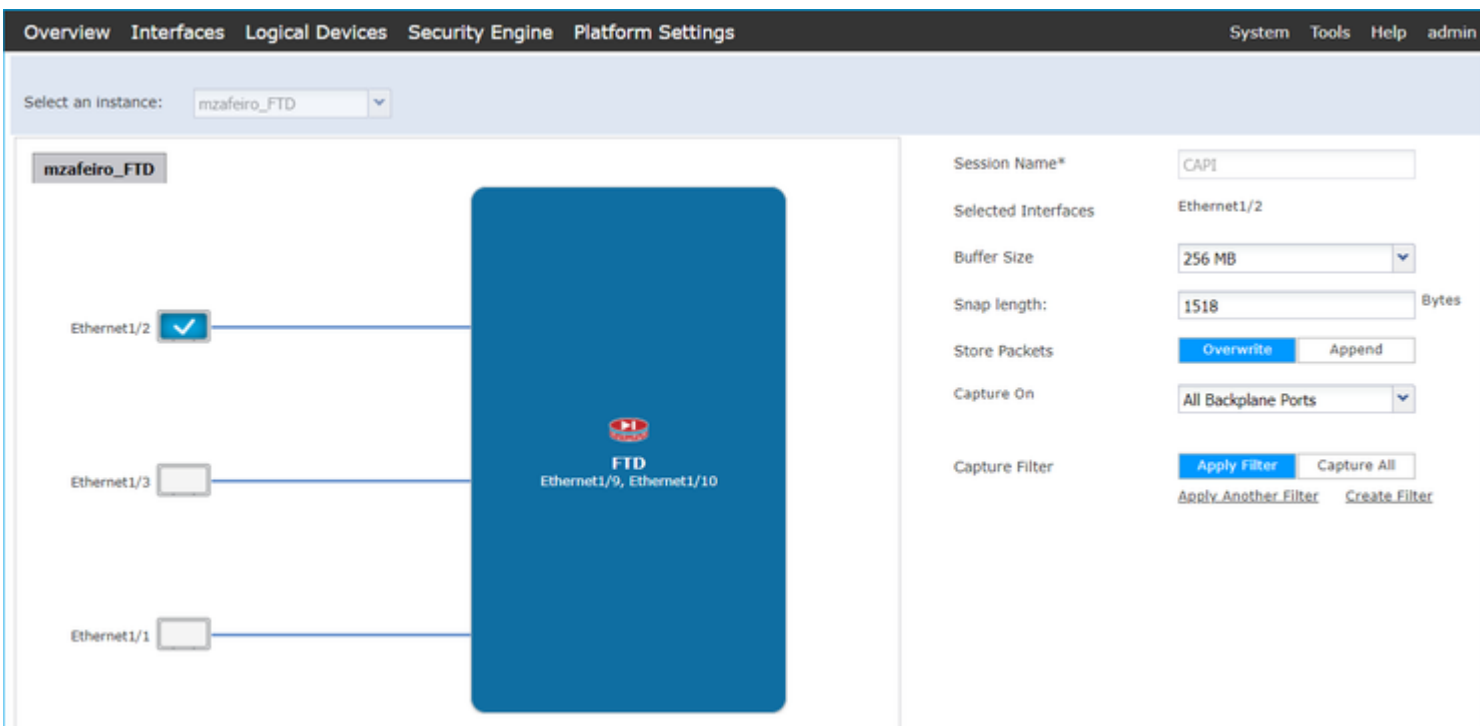
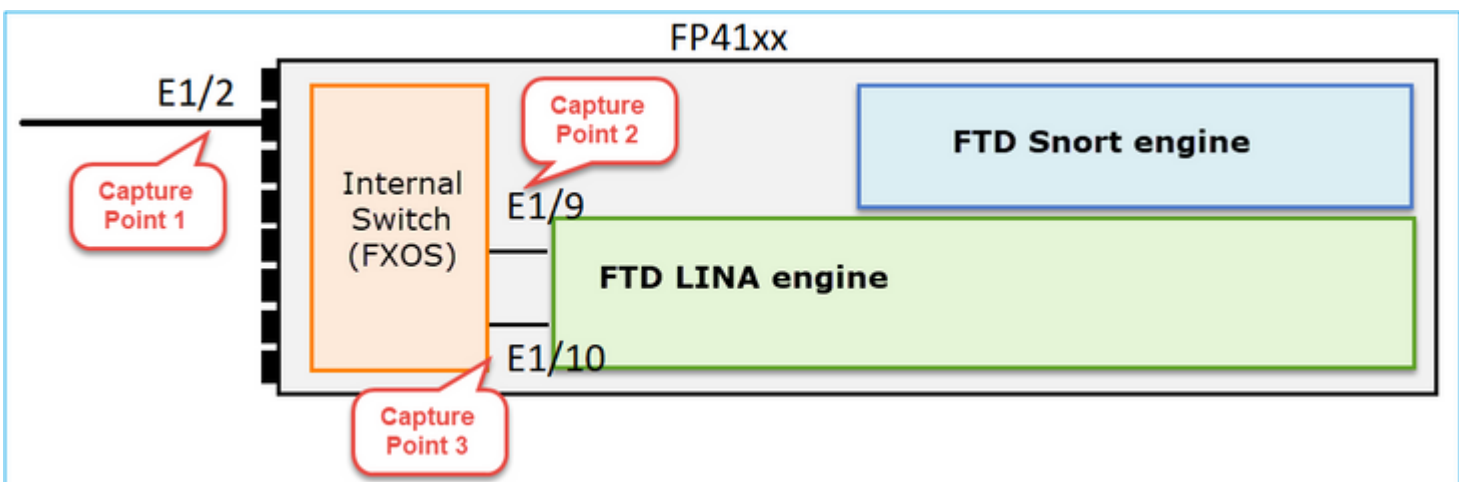
```
N1
```

Wichtigste Punkte:

1. Die Eingangs- und Ausgangsschnittstellen sind identisch (umgekehrt).
2. Die Anzahl der Bytes hat einen signifikant großen Wert (~5 GB).
3. Die Markierung "o" steht für Flow Offload (HW Accelerated Flow). Aus diesem Grund werden bei der FTD-Erfassung keine Pakete angezeigt. Flow Offload wird nur auf 41xx- und 93xx-Plattformen unterstützt. In diesem Fall ist das Gerät ein 41xx.

Maßnahme 2: Nehmen Sie Aufnahmen auf Chassis-Ebene.

Stellen Sie eine Verbindung zum FirePOWER-Chassis-Manager her, und aktivieren Sie die Erfassung an der Eingangsschnittstelle (in diesem Fall E1/2) und an den Backplane-Schnittstellen (E1/9 und E1/10), wie im Bild gezeigt:



Nach einigen Sekunden:

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

Tip: Schließen Sie in Wireshark die mit VN gekennzeichneten Pakete aus, um die Paketduplizierung auf Ebene der physischen Schnittstelle zu vermeiden.

Vorher:

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

Nachher:

CAPI-ethernet-1-2-0.pcap

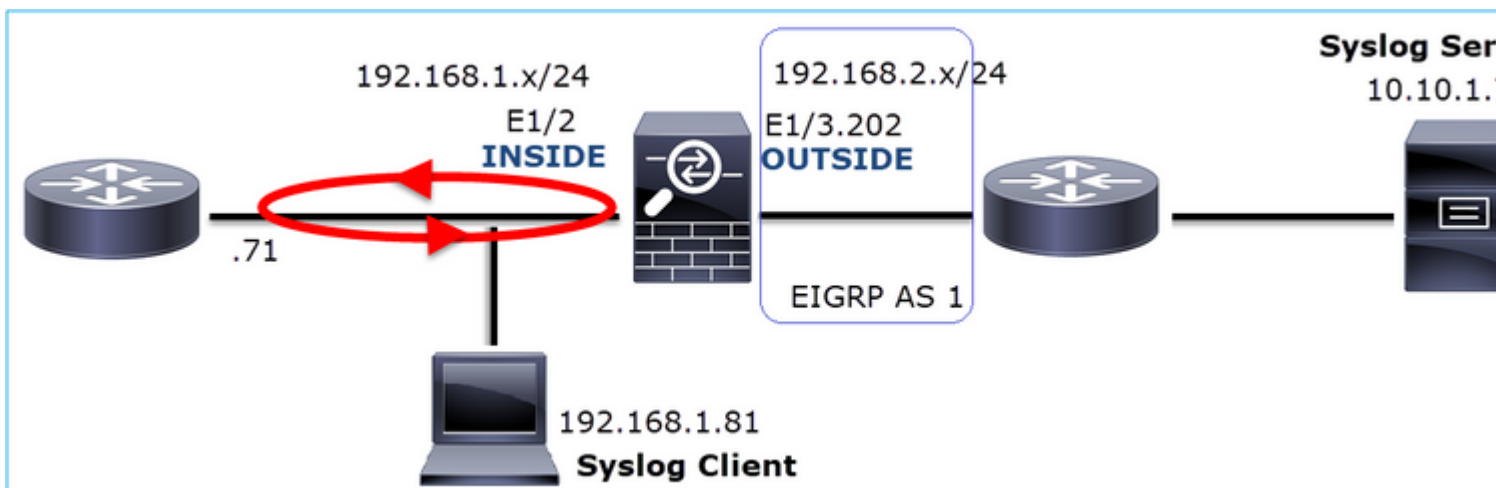
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

syslog && !vntag 1

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

Wichtigste Punkte:

1. Ein Anzeigefilter wird angewendet, um Paketduplikate zu entfernen und nur Syslogs anzuzeigen.
2. Der Unterschied zwischen den Paketen liegt auf der Mikrosekundenebene. Dies weist auf eine sehr hohe Paketrate hin.
3. Der TTL-Wert (Time to Live) nimmt kontinuierlich ab. Zeigt eine Paketschleife an.



Maßnahme 3: Packet-Tracer verwenden.

Da die Pakete die LINA-Engine der Firewall nicht passieren, können Sie keine Live-Verfolgung (Erfassung mit Trace) durchführen. Sie können jedoch ein emuliertes Paket mit Packet Tracer verfolgen:

```
</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
```

Maßnahme 4: Bestätigen Sie die FTD-Weiterleitung.

Prüfen Sie die Routing-Tabelle der Firewall, um festzustellen, ob Routing-Probleme vorliegen:

```
<#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
```

Wichtigste Punkte:

1. Die Route verweist auf die richtige Ausgangsschnittstelle.
2. Die Route wurde vor wenigen Minuten (0:02:37) gelernt.

Maßnahme 5: Bestätigen Sie die Verfügbarkeit der Verbindung.

Überprüfen Sie die Verbindungsverfügbarkeit, um zu sehen, wann diese Verbindung hergestellt wurde:

```
<#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
```

Kernaussage:

1. Die Verbindung wurde vor ca. 4 Minuten hergestellt (vor der EIGRP-Routeninstallation in der Routing-Tabelle).

Maßnahme 6. Löscht die hergestellte Verbindung.

In diesem Fall stimmen die Pakete mit einer bestehenden Verbindung überein und werden an eine falsche Ausgangsschnittstelle weitergeleitet. Dies führt zu einer Schleife. Der Grund hierfür ist die Firewall-Betriebsreihenfolge:

1. Suche nach bestehender Verbindung (hat Vorrang vor der globalen Suche in der Routing-Tabelle).
2. NAT-Suche (Network Address Translation) - Die UN-NAT-Phase (Ziel-NAT) hat Vorrang vor der PBR- und Routensuche.
3. Richtlinienbasiertes Routing
4. Globale Routingtabellen-Suche

Da die Verbindung niemals zu einem Timeout führt (der Syslog-Client sendet fortlaufend Pakete, während die UDP-Konnektivitäts-Zeitüberschreitung 2 Minuten beträgt), muss die Verbindung manuell geleert werden:

```
<#root>
```

```
firepower#
```

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

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

Überprüfen Sie, ob eine neue Verbindung hergestellt wurde:

```

<#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

```

Maßnahme 7. Timeout für Floating-Verbindung konfigurieren

Dies ist die richtige Lösung, um das Problem zu beheben und suboptimales Routing zu vermeiden, insbesondere bei UDP-Datenflüssen. Navigieren Sie zu **Devices (Geräte) > Platform Settings (Plattformeinstellungen) > Timeouts**, und legen Sie den Wert fest:

SMTP Server	H.323	Default	0:0
SNMP	SIP	Default	0:3
SSL	SIP Media	Default	0:0
Syslog	SIP Disconnect:	Default	0:0
Timeouts	SIP Invite	Default	0:0
Time Synchronization	SIP Provisional Media	Default	0:0
UCAPL/CC Compliance	Floating Connection	Custom	0:0
	Xlate-PAT	Default	0:0

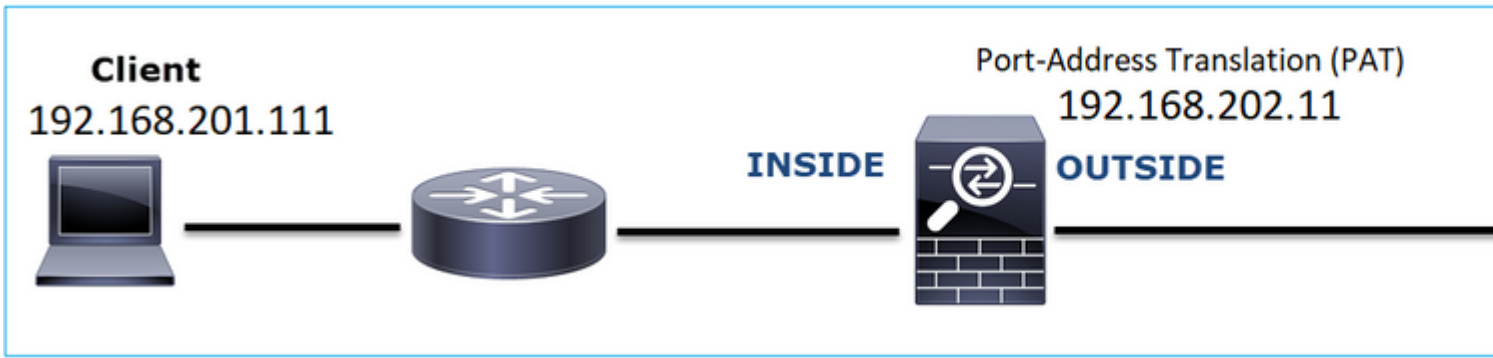
Weitere Details zum Timeout für Floating Conn finden Sie in der Befehlsreferenz:

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

Fall 9: HTTPS-Verbindungsproblem (Szenario 1)

Problembeschreibung: HTTPS-Kommunikation zwischen Client 192.168.201.105 und Server 192.168.202.101 nicht möglich

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Quelle IP: 192.168.201.111

Ziel: 192.168.202.111

Protokoll: TCP 443 (HTTPS)

Erfassungsanalyse

Erfassung auf FTD LINA-Engine aktivieren:

Die bei der OUTSIDE-Erfassung verwendete IP unterscheidet sich aufgrund der Konfiguration für die Port-Adressenumwandlung.

```
<#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
```

Dieses Bild zeigt die Erfassung der NGFW INSIDE-Schnittstelle:

No.	Time	Source	Destination	Protocol	Length	Identification	Info
38	2018-02-01 10:39:35.187887	192.168.201.111	192.168.202.111	TCP	78	0x2f31 (12081)	6666 → 443 [SYN] Seq=2034865631 Win=29200 Len=
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=20348
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
41	2018-02-01 10:39:35.251695	192.168.201.111	192.168.202.111	TLSv1	326	0x2f33 (12083)	Client Hello
42	2018-02-01 10:39:35.252352	192.168.202.111	192.168.201.111	TCP	70	0xefb4 (61364)	443 → 6666 [ACK] Seq=4086514532 Ack=2034865888
43	2018-02-01 10:40:05.317320	192.168.202.111	192.168.201.111	TCP	70	0xd8c3 (55491)	443 → 6666 [RST] Seq=4086514532 Win=8192 Len=6

Wichtigste Punkte:

1. Es gibt einen Drei-Wege-TCP-Handshake.
2. SSL-Aushandlung wird gestartet. Der Client sendet eine Client-Hello-Nachricht.

3. An den Client wird ein TCP-ACK gesendet.
4. Es wird eine TCP-RST an den Client gesendet.

Dieses Bild zeigt die Erfassung der NGFW-OUTSIDE-Schnittstelle.

No.	Time	Source	Destination	Protocol	Length	Identification	Info
33	2018-02-01 10:39:35.188192	192.168.202.11	192.168.202.111	TCP	78	0x2f31 (12081)	15880 → 443 [SYN] Seq=2486930707 Win=29200 Len=0 MSS=1380
34	2018-02-01 10:39:35.188527	192.168.202.111	192.168.202.11	TCP	78	0x0000 (0)	443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=
35	2018-02-01 10:39:35.189214	192.168.202.11	192.168.202.111	TCP	70	0x2f32 (12082)	15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Win=29312
36	2018-02-01 10:39:35.252397	192.168.202.11	192.168.202.111	TLSv1	257	0xcd36 (52534)	Client Hello
37	2018-02-01 10:39:37.274430	192.168.202.11	192.168.202.111	TCP	257	0xb905 (47365)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=248693070
38	2018-02-01 10:39:41.297332	192.168.202.11	192.168.202.111	TCP	257	0x88af (34991)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=248693070
39	2018-02-01 10:39:49.309569	192.168.202.11	192.168.202.111	TCP	257	0xf68a (63114)	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=248693070
40	2018-02-01 10:40:05.317305	192.168.202.11	192.168.202.111	TCP	70	0xd621 (54817)	15880 → 443 [RST] Seq=2486930895 Win=8192 Len=0 TSval=192
41	2018-02-01 10:40:06.790700	192.168.202.111	192.168.202.11	TCP	78	0x0000 (0)	[TCP Retransmission] 443 → 15880 [SYN, ACK] Seq=367440538

Wichtigste Punkte:

1. Es gibt einen Drei-Wege-TCP-Handshake.
2. SSL-Aushandlung wird gestartet. Der Client sendet eine Client-Hello-Nachricht.
3. Es werden TCP-Neuübertragungen von der Firewall an den Server gesendet.
4. Es wird eine TCP-RST an den Server gesendet.

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Machen Sie zusätzliche Aufnahmen.

Eine vom Server durchgeführte Erfassung zeigt, dass der Server die TLS-Client-Hellos mit einer beschädigten TCP-Prüfsumme empfangen hat und diese automatisch verwirft (es gibt keine TCP-RST oder andere Antwortpakete an den Client):

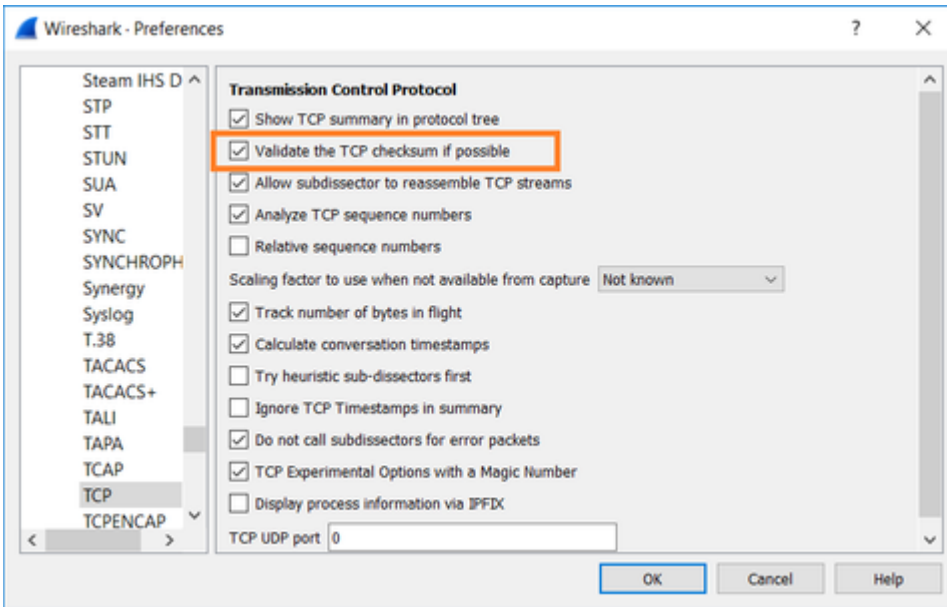
```

21:26:27.133677 IP (tos 0x0, ttl 64, id 52534, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x0c65 (incorrect -> 0x3063), seq 1:188
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
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 d (incorrect -> 0x61fb), seq 1:188
S val 192664224 ecr 0], length 187
21:26:41.189640 IP (tos 0x0, ttl 64, id 63114, offset 0, flags [DF], proto TCP (6), length 239)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [P.], cksum 0x1e9 (incorrect -> 0x42a7), seq 1:188
S val 192672244 ecr 0], length 187
21:26:57.195947 IP (tos 0x0, ttl 64, id 54817, offset 0, flags [DF], proto TCP (6), length 52)
  192.168.202.11.15880 > 192.168.202.111.443: Flags [R], cksum 0x9ee (incorrect -> 0xc2e8), seq 248693
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 36744
ptions [mss 1460,sackOK,TS val 3119647415 ecr 192658158,nop,wscale 7], length 0
^C
154 packets captured
154 packets received by filter

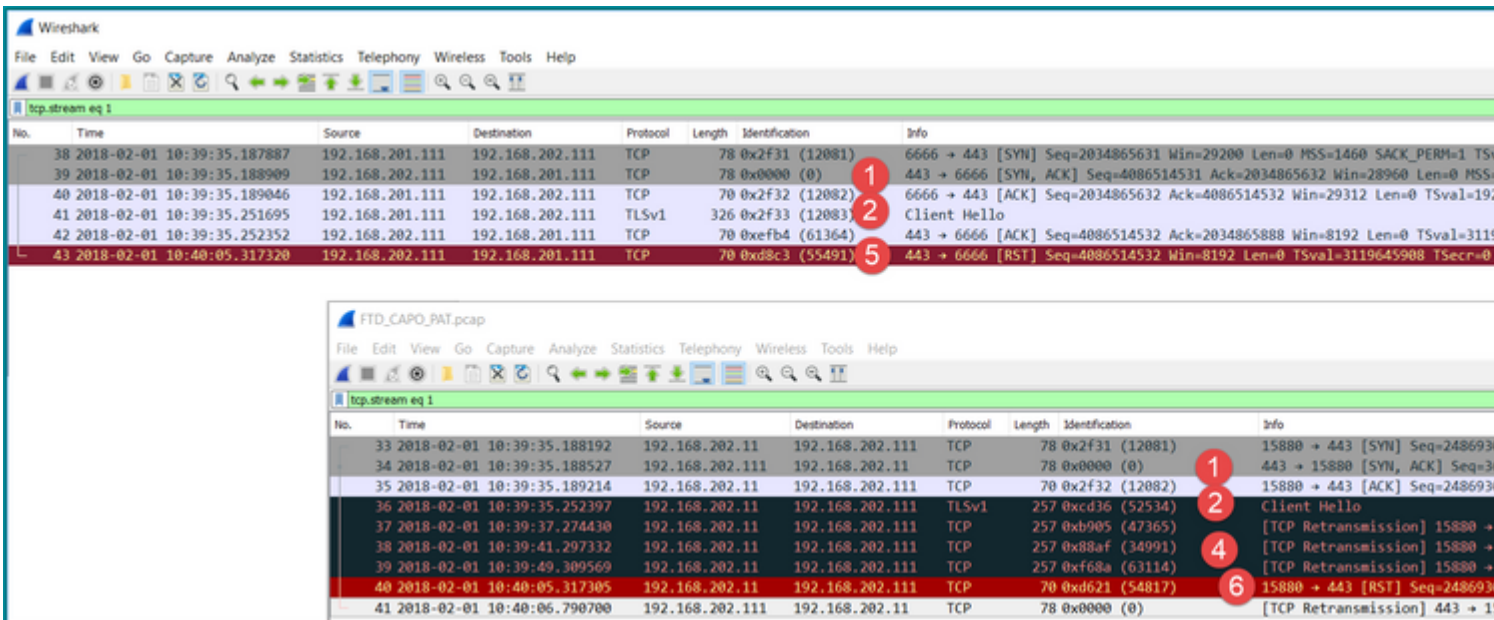
```

Wenn Sie alles zusammenstellen:

In diesem Fall muss, um zu verstehen, in Wireshark die Option **TCP-Prüfsumme validieren**, wenn **möglich**, aktiviert werden. Navigieren Sie zu **Bearbeiten > Voreinstellungen > Protokolle > TCP**, wie im Bild dargestellt.



In diesem Fall ist es hilfreich, die Aufnahmen nebeneinander zu platzieren, um das vollständige Bild zu erhalten:



Wichtigste Punkte:

1. Es gibt einen Drei-Wege-TCP-Handshake. Die IP-IDs sind identisch. Dies bedeutet, dass der Datenfluss nicht von der Firewall bereitgestellt wurde.
2. Ein TLS-Client Hello stammt vom Client mit der IP-ID 12083. Das Paket wird von der Firewall als Proxy weitergeleitet (in diesem Fall wurde die Firewall mit einer TLS-Entschlüsselungsrichtlinie konfiguriert), und die IP-ID wird in 52534 geändert. Außerdem wird die Paket-TCP-Prüfsumme beschädigt (aufgrund eines Softwarefehlers, der später behoben wurde).
3. Die Firewall befindet sich im TCP-Proxy-Modus und sendet eine ACK an den Client (wodurch der Server getäuscht wird).

```

33 2018-02-01 10:39:35.188192 192.168.202.11 192.168.202.111 TCP 78 0x2f31 (12081) 15880 → 443 [SYN] Seq=2486930707 Win=29200 Len=0 MSS=1380 S
34 2018-02-01 10:39:35.188527 192.168.202.111 192.168.202.11 TCP 78 0x0000 (0) 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Min=20
35 2018-02-01 10:39:35.189214 192.168.202.11 192.168.202.111 TCP 70 0x2f32 (12082) 15880 → 443 [ACK] Seq=2486930708 Ack=3674405383 Win=29312 L
36 2018-02-01 10:39:35.252397 192.168.202.11 192.168.202.111 TLSv1 257 0xcd36 (52534) Client Hello

```

```

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

```

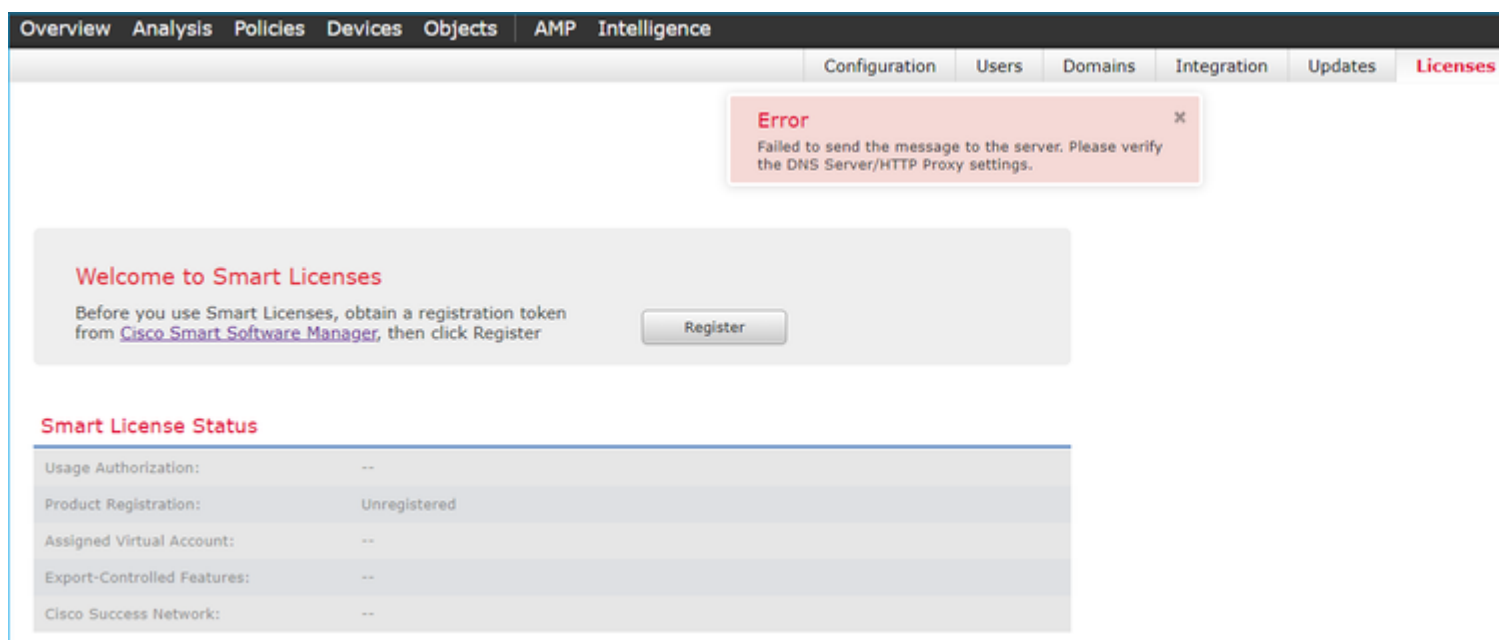
4. Die Firewall empfängt kein TCP-ACK-Paket vom Server und sendet die TLS-Client-Hello-Nachricht erneut. Dies ist wiederum auf den TCP Proxy Modus zurückzuführen, den die Firewall aktiviert hat.
5. Nach ca. 30 Sekunden gibt die Firewall auf und sendet eine TCP-RST an den Client.
6. Die Firewall sendet eine TCP-RST an den Server.

Zur Referenz:

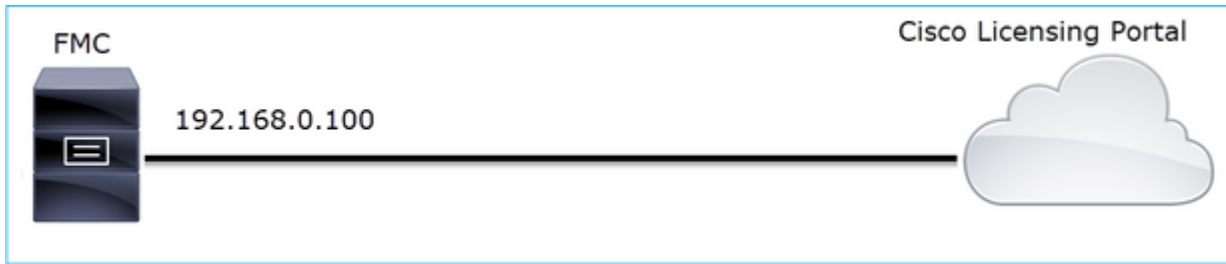
[FirePOWER TLS/SSL-Handshake-Verarbeitung](#)

Fall 10: HTTPS-Verbindungsproblem (Szenario 2)

Problembeschreibung: FMC Smart License Registrierung fehlgeschlagen.



Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

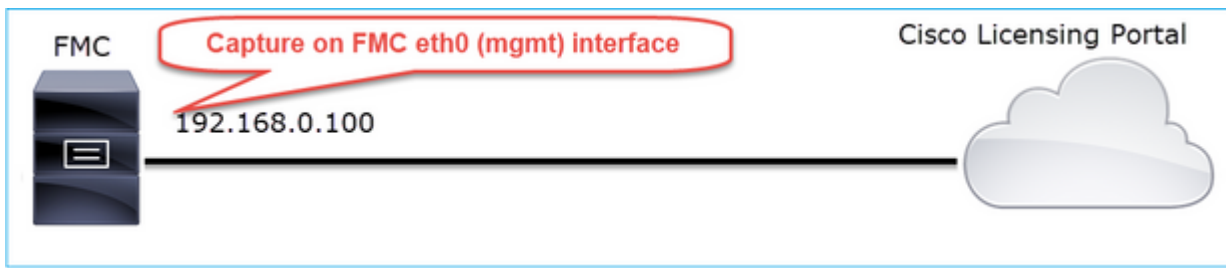
Quelle IP: 192.168.0.100

Ziel: tools.cisco.com

Protokoll: TCP 443 (HTTPS)

Erfassungsanalyse

Aktivieren Sie die Erfassung auf der FMC-Management-Schnittstelle:



Registrieren Sie sich erneut. Sobald die Fehlermeldung angezeigt wird, drücken Sie STRG + C, um die Erfassung zu beenden:

```
<#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
```

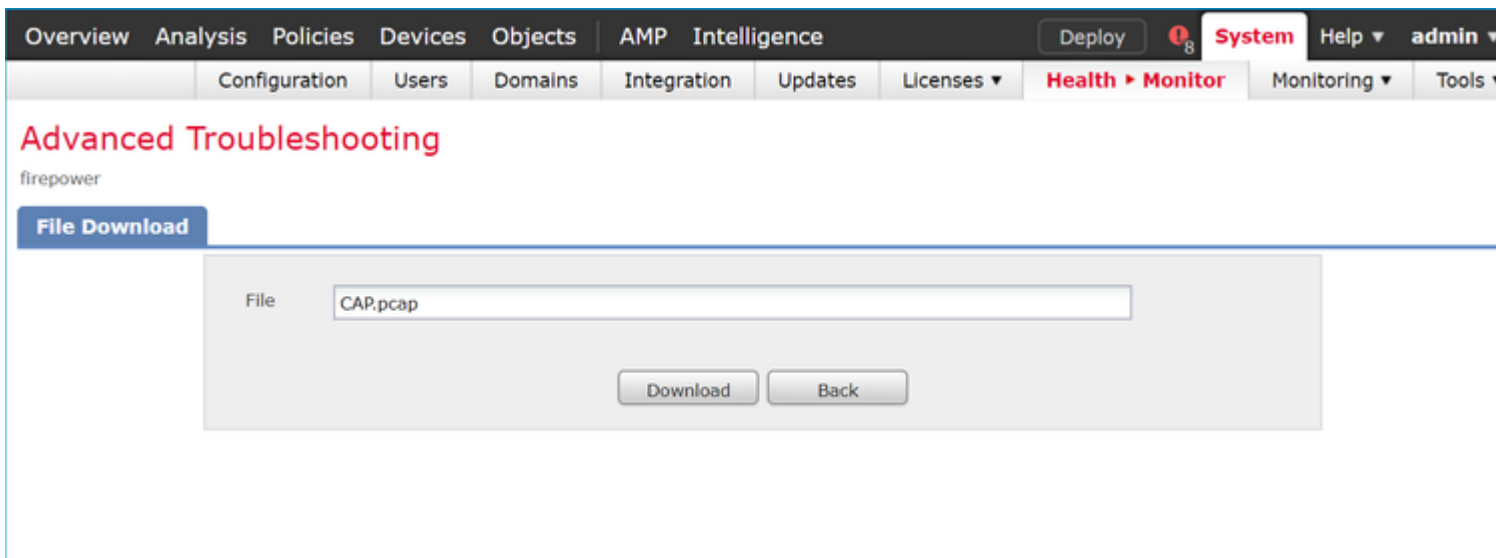
```
<- CTRL-C
```

```
264 packets received by filter
```

```
0 packets dropped by kernel
```

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

Erfassen Sie die Aufzeichnung vom FMC (**System > Health > Monitor (System > Zustand > Monitor)**), wählen Sie das Gerät aus, und wählen Sie **Advanced Troubleshooting (Erweiterte Fehlerbehebung)** aus, wie im Bild gezeigt:



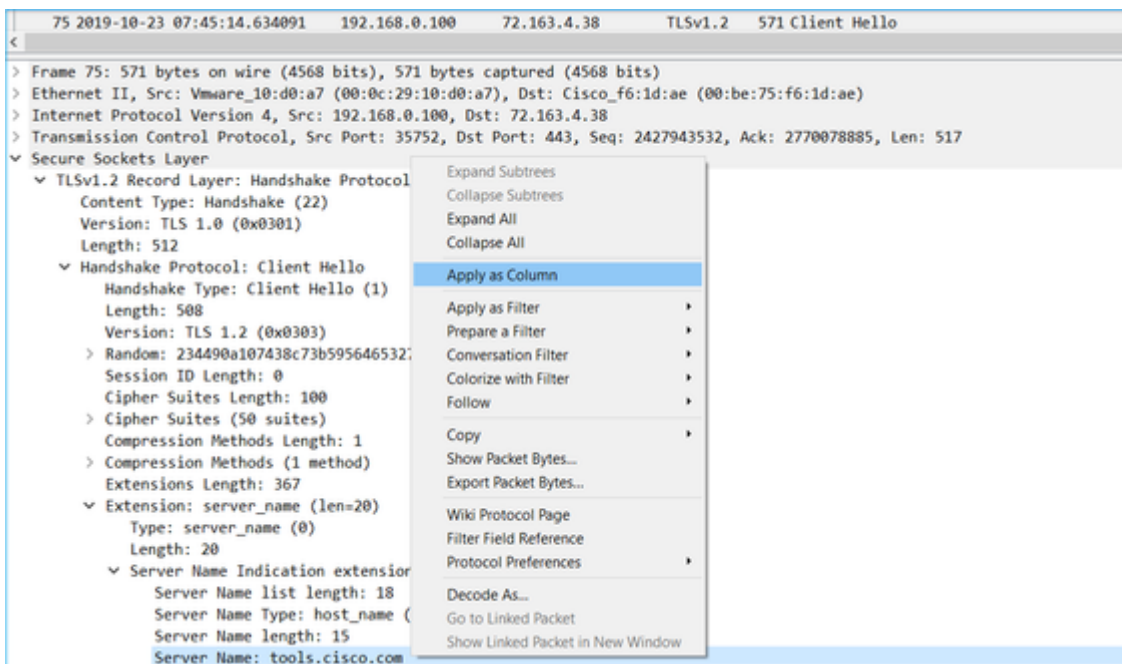
Das Bild zeigt die FMC-Erfassung in Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-23 07:44:59.218797	192.168.0.100	10.229.20.96	TLSv1.2	107	Application Data
2	2019-10-23 07:44:59.220929	10.229.20.96	192.168.0.100	TLSv1.2	123	Application Data
3	2019-10-23 07:44:59.220960	192.168.0.100	10.229.20.96	TCP	54	443 → 64722 [ACK] Seq=13809
4	2019-10-23 07:45:02.215376	192.168.0.100	10.229.20.96	TLSv1.2	107	Application Data
5	2019-10-23 07:45:02.217321	10.229.20.96	192.168.0.100	TLSv1.2	123	Application Data
6	2019-10-23 07:45:02.217336	192.168.0.100	10.229.20.96	TCP	54	443 → 64722 [ACK] Seq=13809
7	2019-10-23 07:45:05.215460	192.168.0.100	10.229.20.96	TLSv1.2	107	Application Data
8	2019-10-23 07:45:05.217331	10.229.20.96	192.168.0.100	TLSv1.2	123	Application Data
9	2019-10-23 07:45:05.217345	192.168.0.100	10.229.20.96	TCP	54	443 → 64722 [ACK] Seq=13809
10	2019-10-23 07:45:06.216584	10.229.20.96	192.168.0.100	TCP	66	64784 → 443 [SYN] Seq=40026
11	2019-10-23 07:45:06.216631	192.168.0.100	10.229.20.96	TCP	66	443 → 64784 [SYN, ACK] Seq=
12	2019-10-23 07:45:06.218550	10.229.20.96	192.168.0.100	TCP	60	64784 → 443 [ACK] Seq=40026
13	2019-10-23 07:45:06.219386	10.229.20.96	192.168.0.100	TLSv1.2	571	Client Hello

Tipp: Um zu überprüfen, ob alle neuen TCP-Sitzungen erfasst wurden, verwenden Sie den `tcp.flags==0x2`-Anzeigefilter in Wireshark. Dadurch werden alle erfassten TCP-SYN-Pakete gefiltert.

No.	Time	Source	Destination	Protocol	Length	Info
10	2019-10-23 07:45:06.216584	10.229.20.96	192.168.0.100	TCP	66	64784 → 443 [SYN] Seq=4002690284 Win=64240 Len=0 MSS=
19	2019-10-23 07:45:06.225743	10.229.20.96	192.168.0.100	TCP	66	64785 → 443 [SYN] Seq=3970528579 Win=64240 Len=0 MSS=
45	2019-10-23 07:45:12.403280	10.229.20.96	192.168.0.100	TCP	66	64790 → 443 [SYN] Seq=442965162 Win=64240 Len=0 MSS=1
51	2019-10-23 07:45:12.409842	10.229.20.96	192.168.0.100	TCP	66	64791 → 443 [SYN] Seq=77539654 Win=64240 Len=0 MSS=13
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74	35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=
108	2019-10-23 07:45:24.969622	192.168.0.100	72.163.4.38	TCP	74	35756 → 443 [SYN] Seq=1993860949 Win=29200 Len=0 MSS=
137	2019-10-23 07:45:35.469403	192.168.0.100	173.37.145.8	TCP	74	58326 → 443 [SYN] Seq=723413997 Win=29200 Len=0 MSS=1
163	2019-10-23 07:45:45.969384	192.168.0.100	173.37.145.8	TCP	74	58330 → 443 [SYN] Seq=2299582550 Win=29200 Len=0 MSS=
192	2019-10-23 07:45:56.468604	192.168.0.100	72.163.4.38	TCP	74	35768 → 443 [SYN] Seq=1199682453 Win=29200 Len=0 MSS=
227	2019-10-23 07:46:07.218984	10.229.20.96	192.168.0.100	TCP	66	64811 → 443 [SYN] Seq=1496581075 Win=64240 Len=0 MSS=
236	2019-10-23 07:46:07.225881	10.229.20.96	192.168.0.100	TCP	66	64812 → 443 [SYN] Seq=563292608 Win=64240 Len=0 MSS=1

Tip: Wenden Sie das Feld **Servername** aus dem SSL Client Hello als Spalte an.



Tip: Wenden Sie diesen Anzeigefilter an, um nur die Client Hello-Nachrichten `ssl.handshake.type == 1` anzuzeigen.

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

Hinweis: Zum Zeitpunkt der Erstellung dieses Dokuments nutzt das Smart Licensing-Portal (tools.cisco.com) die folgenden IP-Adressen: 72.163.4.38, 173.37.145.8

Folgen Sie einem der TCP-Flows (**Folgen > TCP-Stream**), wie im Bild gezeigt.

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

name 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits)
 ethernet II, Src: Vmware_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
 Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38
 Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 512
 Secure Sockets Layer

▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
 Content Type: Handshake (22)
 Version: TLS 1.0 (0x0301)
 Length: 512

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

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74		35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=
73	2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [SYN, ACK] Seq=2770078884 Ack=2427943532
74	2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427943532 Ack=2770078885 Win=
75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
76	2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=
77	2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150		Server Hello
78	2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=
79	2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1384		443 → 35752 [PSH, ACK] Seq=2770078981 Ack=2427944049
80	2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=
81	2019-10-23 07:45:14.966877	72.163.4.38	192.168.0.100	TLSv1.2	155		Certificate
82	2019-10-23 07:45:14.966887	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080412 Win=
83	2019-10-23 07:45:14.966915	72.163.4.38	192.168.0.100	TLSv1.2	63		Server Hello Done
84	2019-10-23 07:45:14.966925	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080421 Win=
85	2019-10-23 07:45:14.967114	192.168.0.100	72.163.4.38	TLSv1.2	61		Alert (Level: Fatal, Description: Unknown CA)
86	2019-10-23 07:45:14.967261	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [RST, ACK] Seq=2427944056 Ack=2770080421 Win=
87	2019-10-23 07:45:14.967382	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770080421 Ack=2427944056 Win=
88	2019-10-23 07:45:14.967398	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [RST] Seq=2427944056 Win=0 Len=0

> Frame 75: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits)
 > Ethernet II, Src: Vmware_10:d0:a7 (00:0c:29:10:d0:a7), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae)
 > Internet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38
 > Transmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Len: 512
 ▼ Secure Sockets Layer

▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
 Content Type: Handshake (22)
 Version: TLS 1.0 (0x0301)
 Length: 512

▼ Handshake Protocol: Client Hello
 Handshake Type: Client Hello (1)
 Length: 508
 Version: TLS 1.2 (0x0303)
 > Random: 234490a107438c73b59564653271c7c09fbbb7ac16897184...
 Session ID Length: 0
 Cipher Suites Length: 100
 > Cipher Suites (50 suites)

Wichtigste Punkte:

1. Es gibt einen Drei-Wege-TCP-Handshake.
2. Der Client (FMC) sendet eine SSL Client Hello-Nachricht an das Smart Licensing-Portal.
3. Die SSL-Sitzungs-ID lautet 0. Dies bedeutet, dass es sich nicht um eine wiederaufgenommene Sitzung handelt.
4. Der Zielservers antwortet mit der "Server Hello"-, "Certificate"- und "Server Hello Done"-Nachricht.
5. Der Client sendet eine schwerwiegende SSL-Warnung, die eine "unbekannte Zertifizierungsstelle" betrifft.
6. Der Client sendet eine TCP-RST, um die Sitzung zu schließen.
7. Die gesamte TCP-Sitzungsdauer (von der Einrichtung bis zum Abschluss) betrug ca. 0,5 Sekunden.

Wählen Sie das **Serverzertifikat aus**, und erweitern Sie das **Ausstellerfeld**, um den commonName anzuzeigen. In diesem Fall zeigt der Common Name ein Gerät, das Man-in-the-Middle (MITM) unterstützt.

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
72	2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74		35752 → 443 [SYN] Seq=2427943531 Wi
73	2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [SYN, ACK] Seq=27700788
74	2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427943532 Ac
75	2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
76	2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770078885 Ac
77	2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150		Server Hello
78	2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ac
79	2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1384		443 → 35752 [PSH, ACK] Seq=27700789
80	2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ac
81	2019-10-23 07:45:14.966872	72.163.4.38	192.168.0.100	TLSv1.2	155		Certificate


```

Length: 1426
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1422
    Certificates Length: 1419
  Certificates (1419 bytes)
    Certificate Length: 1416
  Certificate: 308205843082046ca003020102020d00aa23af5d607e0000... (id-at-commonName=tools.cisco.com,id-at-organizationName=Cisco Sy
    signedCertificate
      version: v3 (2)
      serialNumber: 0x00aa23af5d607e00002f423880
      signature (sha256WithRSAEncryption)
        issuer: rdnSequence (0)
          rdnSequence: 3 items (id-at-commonName=FTD4100_MITM,id-at-organizationalUnitName=FTD_OU,id-at-organizationName=FTD_0)
            RDNSquence item: 1 item (id-at-organizationName=FTD_0)
            RDNSquence item: 1 item (id-at-organizationalUnitName=FTD_OU)
            RDNSquence item: 1 item (id-at-commonName=FTD4100_MITM)
          validity
          subject: rdnSequence (0)
          subjectPublicKeyInfo
        extensions: 6 items
  
```

Dies wird in der folgenden Abbildung dargestellt:

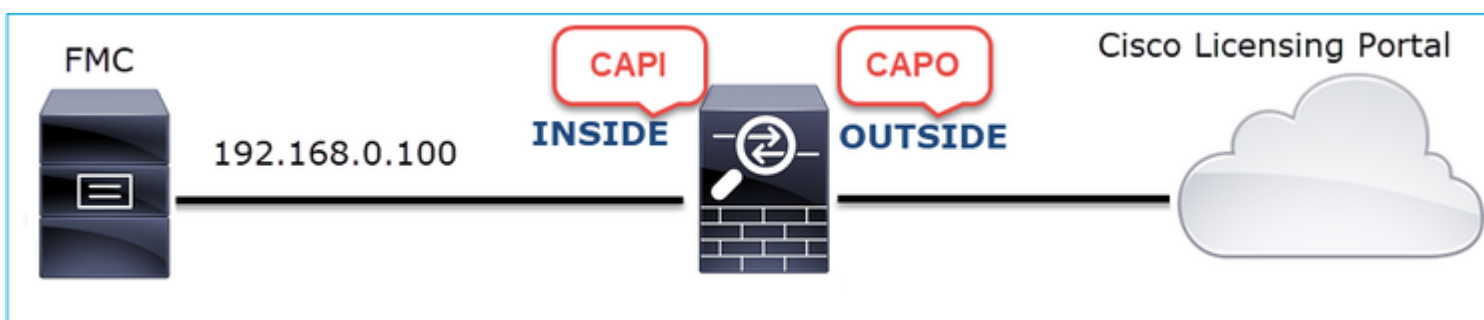


Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Machen Sie zusätzliche Aufnahmen.

Erfassung auf dem Transport-Firewall-Gerät:



CAPI zeigt:

tcp.stream eq 57

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1221	2019-10-22 17:49:03.212681	192.168.0.100	173.37.145.8	TCP	74		39924 → 443 [SYN] Seq=42
1222	2019-10-22 17:49:03.379023	173.37.145.8	192.168.0.100	TCP	58		443 → 39924 [SYN, ACK] S
1223	2019-10-22 17:49:03.379298	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=42
1224	2019-10-22 17:49:03.380336	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
1225	2019-10-22 17:49:03.380732	173.37.145.8	192.168.0.100	TCP	54		443 → 39924 [ACK] Seq=23
1226	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TLSv1.2	150		Server Hello
1227	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TCP	1384		443 → 39924 [PSH, ACK] S
1228	2019-10-22 17:49:03.710092	173.37.145.8	192.168.0.100	TLSv1.2	155		Certificate
1229	2019-10-22 17:49:03.710107	173.37.145.8	192.168.0.100	TLSv1.2	63		Server Hello Done
1230	2019-10-22 17:49:03.710412	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=42
1231	2019-10-22 17:49:03.710519	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=42
1232	2019-10-22 17:49:03.710519	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=42
1233	2019-10-22 17:49:03.710534	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [ACK] Seq=42
1234	2019-10-22 17:49:03.710626	192.168.0.100	173.37.145.8	TLSv1.2	61		Alert (Level: Fatal, Des
1235	2019-10-22 17:49:03.710641	173.37.145.8	192.168.0.100	TCP	54		443 → 39924 [ACK] Seq=23
1236	2019-10-22 17:49:03.710748	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [RST, ACK] S
1237	2019-10-22 17:49:03.710870	192.168.0.100	173.37.145.8	TCP	54		39924 → 443 [RST] Seq=42

Length: 1426

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

CAPO zeigt an:

No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1169	2019-10-22 17:49:03.212849	192.168.0.100	173.37.145.8	TCP	78		39924 → 443 [SYN] Seq=623942
1170	2019-10-22 17:49:03.378962	173.37.145.8	192.168.0.100	TCP	62		443 → 39924 [SYN, ACK] Seq=4
1171	2019-10-22 17:49:03.379329	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942
1172	2019-10-22 17:49:03.380793	192.168.0.100	173.37.145.8	TLSv1.2	512	tools.cisco.com	Client Hello
1173	2019-10-22 17:49:03.545748	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4
1174	2019-10-22 17:49:03.545809	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4
1175	2019-10-22 17:49:03.545824	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942
1176	2019-10-22 17:49:03.545915	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4
1177	2019-10-22 17:49:03.545961	173.37.145.8	192.168.0.100	TCP	1388		443 → 39924 [PSH, ACK] Seq=4
1178	2019-10-22 17:49:03.545961	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [ACK] Seq=623942
1179	2019-10-22 17:49:03.709420	173.37.145.8	192.168.0.100	TLSv1.2	82		Server Hello, Certificate, S
1180	2019-10-22 17:49:03.710687	192.168.0.100	173.37.145.8	TLSv1.2	65		Alert (Level: Fatal, Descrip
1181	2019-10-22 17:49:03.710885	192.168.0.100	173.37.145.8	TCP	58		39924 → 443 [FIN, PSH, ACK]
1182	2019-10-22 17:49:03.874542	173.37.145.8	192.168.0.100	TCP	58		443 → 39924 [RST, ACK] Seq=4


```

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

Diese Erfassungen belegen, dass die Transit-Firewall das Serverzertifikat (MITM) ändert.

Maßnahme 2: Überprüfen der Geräteprotokolle

Sie können das FMC TS-Paket wie in diesem Dokument beschrieben sammeln:

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

In diesem Fall zeigt die Datei `/dir-archives/var-log/process_stdout.log` folgende Meldungen an:

```

<#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[49]
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_iss
cert issue checking, ret 60, url "https://tools.cisco.com/its/
  
```

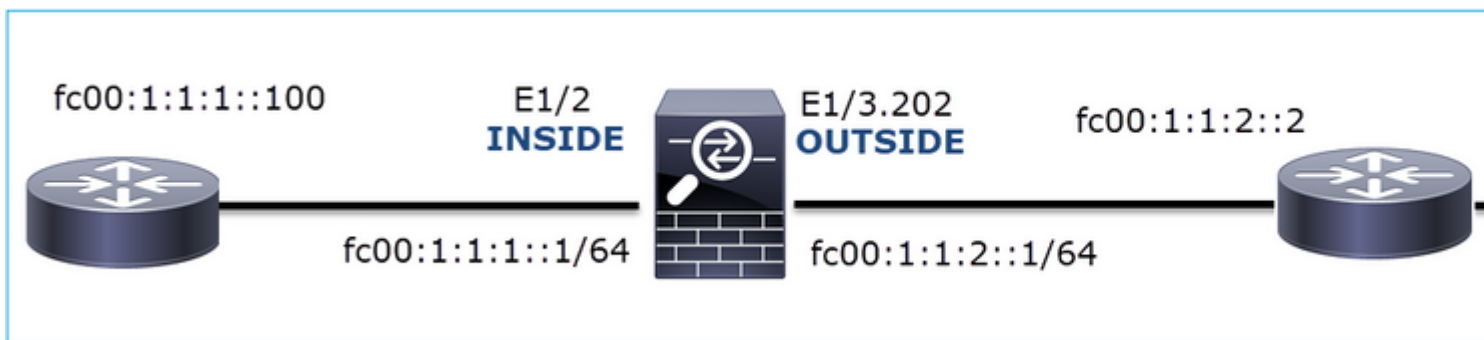
Empfohlene Lösung

Deaktivieren Sie den MITM für den jeweiligen Fluss, damit FMC sich erfolgreich bei der Smart Licensing-Cloud registrieren kann.

Fall 11: IPv6-Verbindungsproblem

Problembeschreibung: Interne Hosts (die sich hinter der INSIDE-Schnittstelle der Firewall befinden) können nicht mit externen Hosts (Hosts, die sich hinter der OUTSIDE-Schnittstelle der Firewall befinden) kommunizieren.

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Src-IP: `fc00:1:1:1::100`

Ziel-IP: `fc00:1:1:2::2`

Protokoll: Beliebig

Erfassungsanalyse

Aktivieren Sie Aufnahmen auf FTD LINA-Engine.

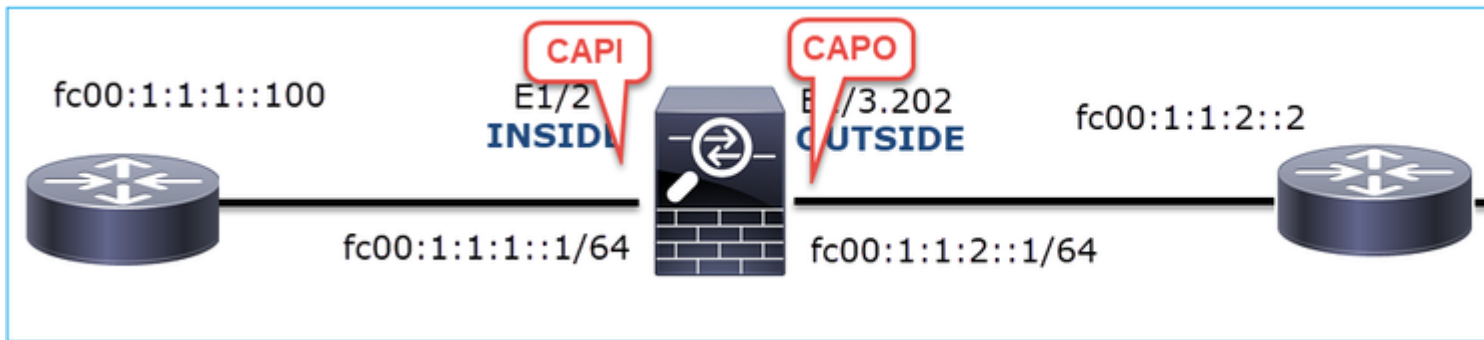
```
<#root>
```

```
firepower#
```

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

```
firepower#
```

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



Erfassungen - Nicht-Funktionsszenario

Diese Erfassungen wurden parallel zu einem ICMP-Verbindungstest von IP fc00:1:1:1::100 (interner Router) zu IP fc00:1:1:2::2 (Upstream-Router) durchgeführt.

Die Erfassung auf der Firewall INSIDE-Schnittstelle enthält:

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
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
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
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
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
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::1
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::1
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
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
10	2019-10-24 13:02:17.062725	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1dae	ICMPv6	86	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8
11	2019-10-24 13:02:17.062862	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	78	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8
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
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:fef6:1dae

Wichtigste Punkte:

1. Der Router sendet eine IPv6 Neighbor Solicitation-Nachricht und fragt nach der MAC-Adresse des Upstream-Geräts (IP fc00:1:1:1::1).
2. Die Firewall antwortet mit einer IPv6 Neighbor Advertisement.
3. Der Router sendet eine ICMP-Echoanfrage.
4. Die Firewall sendet eine IPv6 Neighbor Solicitation-Nachricht und fragt die MAC-Adresse des Downstream-Geräts ab (fc00:1:1:1::100).
5. Der Router antwortet mit einer IPv6 Neighbor Advertisement.
6. Der Router sendet zusätzliche IPv6-ICMP-Echoanfragen.

Die Erfassung auf der OUTSIDE-Schnittstelle der Firewall umfasst:

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
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
3	2019-10-24 13:02:08.997995	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	18	Echo (ping) request id=0x160d
4	2019-10-24 13:02:09.001815	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
5	2019-10-24 13:02:10.025938	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
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
7	2019-10-24 13:02:11.050015	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
8	2019-10-24 13:02:12.066082	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	90	Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8
9	2019-10-24 13:02:12.066234	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	82	Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8
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
11	2019-10-24 13:02:13.002105	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
12	2019-10-24 13:02:14.090251	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
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
14	2019-10-24 13:02:15.178350	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
15	2019-10-24 13:02:17.059963	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6	90	Neighbor Solicitation for fe80::2be:75ff:fef6:1d8e
16	2019-10-24 13:02:17.062512	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6	82	Neighbor Advertisement fe80::2be:75ff:fef6:1d8e

Wichtigste Punkte:

1. Die Firewall sendet eine IPv6 Neighbor Solicitation-Nachricht, in der die MAC-Adresse des Upstream-Geräts angefordert wird (IP fc00:1:1:2::2).
2. Der Router antwortet mit einer IPv6 Neighbor Advertisement.
3. Die Firewall sendet eine IPv6-ICMP-Echoanfrage.
4. Das Upstream-Gerät (Router fc00:1:1:2::2) sendet eine IPv6 Neighbor Solicitation-Nachricht, in der die MAC-Adresse der IPv6-Adresse fc00:1:1:1::100 angefordert wird.
5. Die Firewall sendet eine zusätzliche IPv6-ICMP-Echoanfrage.
6. Der Upstream-Router sendet eine zusätzliche IPv6 Neighbor Solicitation-Nachricht, in der die MAC-Adresse der IPv6-Adresse fc00:1:1:1::100 angefordert wird.

Punkt 4 ist sehr interessant. Normalerweise fragt der Upstream-Router nach der MAC-Adresse der OUTSIDE-Schnittstelle der Firewall (fc00:1:1:2::2), aber stattdessen nach der fc00:1:1:1::100. Dies ist ein Hinweis auf eine fehlerhafte Konfiguration.

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Überprüfen Sie die IPv6 Neighbor Table.

Die IPv6-Nachbartabelle für die Firewall ist ordnungsgemäß ausgefüllt.

```
<#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
```

Maßnahme 2: Überprüfen der IPv6-Konfiguration

Dies ist die Firewall-Konfiguration.

```
<#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

```

Die Konfiguration des Upstream-Geräts zeigt die fehlerhafte Konfiguration:

```

<#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

```

Erfassungen - Funktionsszenario

Durch die Änderung der Subnetzmaske (von /48 auf /64) wurde das Problem behoben. Dies ist die CAPI-Erfassung im funktionalen Szenario.

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 fc00:1:1:2::2
2	2019-10-24 15:17:20.677989	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6	86	Neighbor Advertisement fc00:1:1:2::2
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=1
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=2
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=3
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=1
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=4
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=2
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=5
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=3

Kernaussage:

1. Der Router sendet eine IPv6 Neighbor Solicitation-Nachricht, in der er nach der MAC-Adresse des

- Upstream-Geräts fragt (IP fc00:1:1:1::1).
2. Die Firewall antwortet mit einer IPv6 Neighbor Advertisement.
 3. Der Router sendet ICMP-Echoanfragen und erhält Echoantworten.

CAPO-Inhalt:

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-10-24 15:17:20.678645	fe80::2be:75ff:fe...	ff02::1:ff00:2	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::2
2	2019-10-24 15:17:20.681818	fc00:1:1:2::2	fe80::2be:75ff:fe...	ICMPv6	90	Neighbor Advertisement fc00:1:1:2::2
3	2019-10-24 15:17:22.674342	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118	Echo (ping) request id=0x097e, seq=1
4	2019-10-24 15:17:22.677943	fc00:1:1:2::2	ff02::1:ff00:1	ICMPv6	90	Neighbor Solicitation for fc00:1:1:2::1
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
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
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
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
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
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
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
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

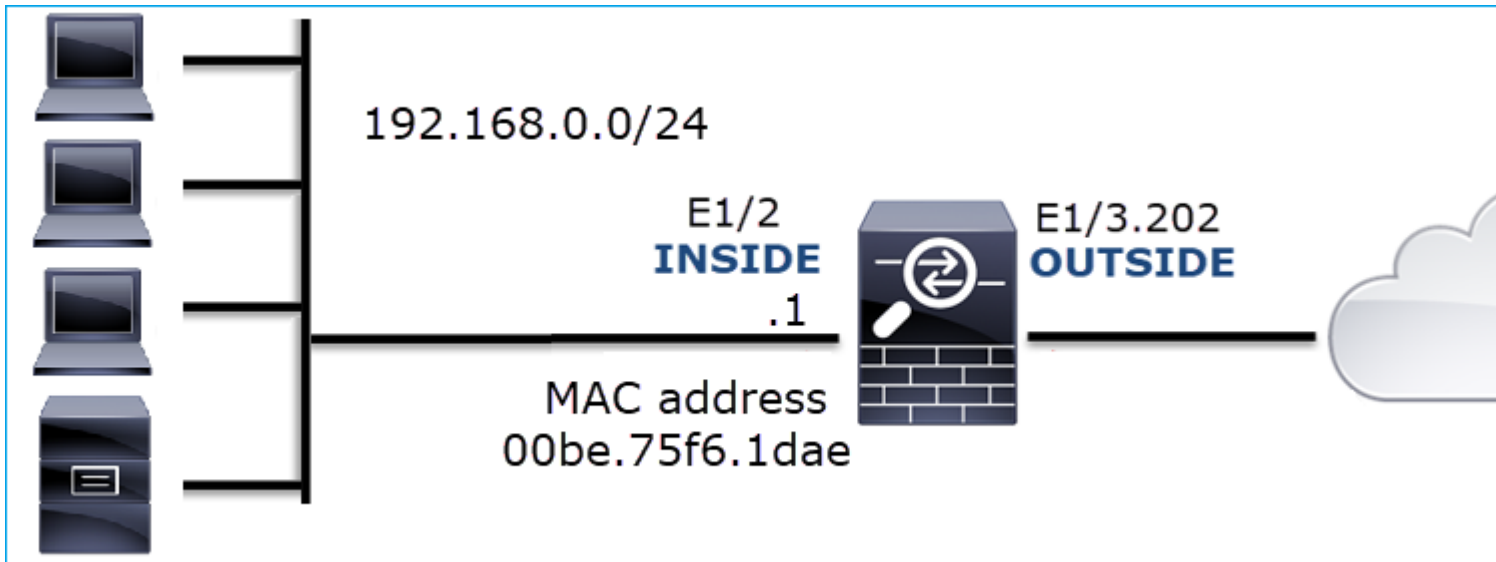
Wichtigste Punkte:

1. Die Firewall sendet eine IPv6 Neighbor Solicitation-Nachricht, in der die MAC-Adresse des Upstream-Geräts angefordert wird (IP fc00:1:1:2::2).
2. Die Firewall antwortet mit einer IPv6 Neighbor Advertisement.
3. Die Firewall sendet eine ICMP-Echoanfrage.
4. Der Router sendet eine IPv6 Neighbor Solicitation-Nachricht, in der er nach der MAC-Adresse des Downstream-Geräts fragt (IP fc00:1:1:1::1).
5. Die Firewall antwortet mit einer IPv6 Neighbor Advertisement.
6. Die Firewall sendet ICMP-Echoanfragen und erhält Echoantworten.

Fall 12: Intermittierendes Verbindungsproblem (ARP Poisoning)

Problembeschreibung: Interne Hosts (192.168.0.x/24) haben zeitweilige Verbindungsprobleme mit Hosts im gleichen Subnetz

Dieses Bild zeigt die Topologie:



Betroffener Datenfluss:

Quelle IP: 192.168.0.x/24

Ziel-IP: 192.168.0.x/24

Protokoll: Beliebig

Der ARP-Cache eines internen Hosts ist anscheinend beschädigt:

```

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

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

C:\Users\mzafeiro1>

```

Erfassungsanalyse

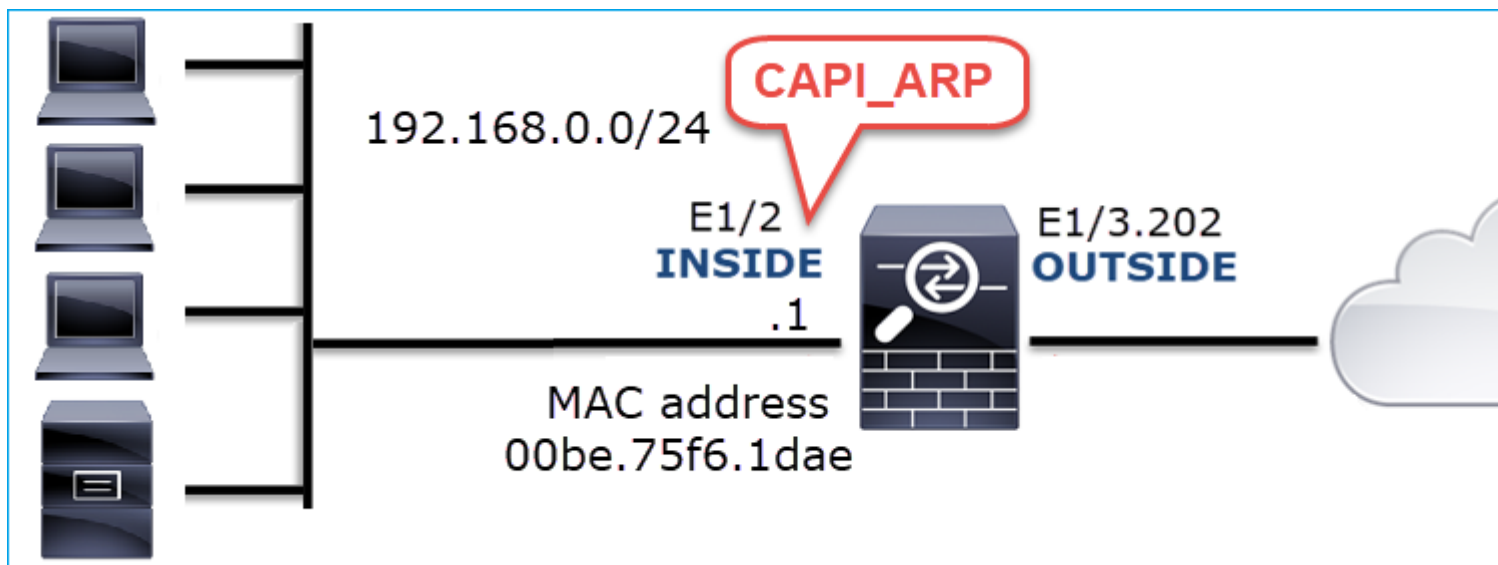
Erfassung auf FTD LINA-Engine aktivieren

Diese Erfassung erfasst nur ARP-Pakete an der INSIDE-Schnittstelle:

```
<#root>
```

```
firepower#
```

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



Erfassungen - Nicht-Funktionsszenario:

Die Erfassung auf der Firewall INSIDE-Schnittstelle enthält

```
[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	1	60 Who has 192.168.0.1
5	2019-10-25 10:01:55.17969	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.23
35	2019-10-25 10:02:13.050397	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
36	2019-10-25 10:02:13.050488	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.24
47	2019-10-25 10:02:19.284683	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
48	2019-10-25 10:02:19.284775	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.25
61	2019-10-25 10:02:25.779821	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
62	2019-10-25 10:02:25.779912	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.26
76	2019-10-25 10:02:31.978175	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
77	2019-10-25 10:02:31.978251	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.27
97	2019-10-25 10:02:38.666515	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
98	2019-10-25 10:02:38.666606	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.28
121	2019-10-25 10:02:47.384074	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
122	2019-10-25 10:02:47.384150	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.29
137	2019-10-25 10:02:53.539995	Vmware_2c:9b:a7	Broadcast	ARP	1	60 Who has 192.168.0.1
138	2019-10-25 10:02:53.540087	Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP	1	42 192.168.0.30

Wichtigste Punkte:

1. Die Firewall empfängt verschiedene ARP-Anfragen für IPs innerhalb des Netzwerks 192.168.0.x/24
2. Die Firewall beantwortet alle Anfragen (Proxy-ARP) mit einer eigenen MAC-Adresse

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Maßnahme 1: Überprüfen der NAT-Konfiguration

In Bezug auf die NAT-Konfiguration gibt es Fälle, in denen das **no-proxy-arp**-Schlüsselwort das frühere Verhalten verhindern kann:

```
<#root>
```

```
firepower#
```

```
show run nat
```

```
nat (INSIDE,OUTSIDE) source static NET_1.1.1.0 NET_2.2.2.0 destination static NET_192.168.0.0 NET_4.4.4.4
```

```
no-proxy-arp
```

Maßnahme 2: Deaktivieren Sie die Proxy-ARP-Funktion an der Firewall-Schnittstelle.

Wenn das Schlüsselwort "no-proxy-arp" das Problem nicht löst, versuchen Sie, den Proxy-ARP auf der Schnittstelle selbst zu deaktivieren. Bei FTD müssen Sie zum Zeitpunkt der Erstellung dieses Dokuments FlexConfig verwenden und den Befehl bereitstellen (geben Sie den entsprechenden Schnittstellennamen an).

```
sysopt noproxyarp INSIDE
```

Fall 13: Identifizieren von SNMP-Objektbezeichnern (OIDs), die CPU-Hogs verursachen

Dieser Fall zeigt, wie bestimmte SNMP OIDs für das Speicher-Polling auf Basis der Analyse von SNMP-Paketerfassungen der Version 3 (SNMPv3) als Ursache von CPU-Hogs (Leistungsproblemen) identifiziert wurden.

Problembeschreibung: Die Überlastung der Datenschnittstellen nimmt stetig zu. Weitere Untersuchungen haben ergeben, dass es auch CPU-Hogs gibt (verursacht durch den SNMP-Prozess), die die Ursache für die Schnittstellenüberläufe sind.

Der nächste Schritt beim Fehlerbehebungsprozess bestand darin, die Ursache der CPU-Probleme zu identifizieren, die durch den SNMP-Prozess verursacht wurden. Insbesondere sollte der Umfang des Problems eingegrenzt werden, um die SNMP-Objektbezeichner (OID) zu identifizieren, die beim Abfragen möglicherweise zu CPU-Problemen führen können.

Derzeit bietet die FTD LINA-Engine keinen "show"-Befehl für SNMP-OIDs, die in Echtzeit abgefragt werden.

Die Liste der SNMP OIDs für das Polling kann vom SNMP-Überwachungstool abgerufen werden. In diesem Fall gab es jedoch die folgenden vorbeugenden Faktoren:

- Der FTD-Administrator hatte keinen Zugriff auf das SNMP-Überwachungstool.
- SNMP-Version 3 mit Authentifizierung und Datenverschlüsselung wurde auf FTD konfiguriert.

Erfassungsanalyse

Da der FTD-Administrator über die Anmeldeinformationen für die SNMP-Authentifizierung Version 3 und die Datenverschlüsselung verfügte, wurde dieser Aktionsplan vorgeschlagen:

1. SNMP-Paketerfassung übernehmen
2. Speichern Sie die Aufzeichnungen, und verwenden Sie die Wireshark SNMP-Protokolleinstellungen, um die SNMP-Anmeldeinformationen der Version 3 zum Entschlüsseln der SNMP-Pakete der Version 3 anzugeben. Die entschlüsselten Erfassungen werden für die Analyse und den Abruf von SNMP OIDs verwendet.

Konfigurieren Sie die SNMP-Paketerfassung auf der Schnittstelle, die in der snmp-server-Hostkonfiguration verwendet wird:

```
<#root>
firepower#
show run snmp-server | include host
snmp-server host management 192.168.10.10 version 3 netmonv3

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

firepower#
capture capsnpmp interface management buffer 10000000 match udp host 192.168.10.10 host 192.168.5.254 eq

firepower#
show capture capsnpmp

capture capsnpmp type raw-data buffer 10000000 interface outside [Capturing -
9512
bytes]
match udp host 192.168.10.10 host 192.168.5.254 eq snmp
```

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


```

<[Destination Host: 192.168.5.254]>
<[Source or Destination Host: 192.168.5.254]>
> User Datagram Protocol, Src Port: 65484, Dst Port: 161
< Simple Network Management Protocol
  msgVersion: snmpv3 (3)
  > msgGlobalData
  > msgAuthoritativeEngineID: 80000009fe1c6dad4930a00ef1fec2301621a4158bfc1f40...
  msgAuthoritativeEngineBoots: 0
  msgAuthoritativeEngineTime: 0
  msgUserName: netmonv3
  msgAuthenticationParameters: ff5176f5973c30b62ffc11b8
  msgPrivacyParameters: 000040e100003196
  < msgData: encryptedPDU (1)
    encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703...
  
```

Wichtigste Punkte:

1. SNMP-Quell- und Zieladressen/-ports.
2. Die SNMP-Protokoll-PDU konnte nicht dekodiert werden, da der privKey für Wireshark unbekannt ist.
3. Der Wert der verschlüsselten PDU-Grundeinheit.

Empfohlene Maßnahmen

Mit den in diesem Abschnitt aufgeführten Maßnahmen soll das Problem weiter eingegrenzt werden.

Aktion 1. Entschlüsseln der SNMP-Erfassungen

Speichern Sie die Aufzeichnungen, und bearbeiten Sie die Einstellungen des Wireshark-SNMP-Protokolls, um die SNMP-Anmeldeinformationen der Version 3 zum Entschlüsseln der Pakete anzugeben.

<#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

Öffnen Sie die Erfassungsdatei in Wireshark, wählen Sie ein SNMP-Paket aus, und navigieren Sie zu **Protokolleinstellungen > Benutzertabelle**, wie im Bild gezeigt:

The screenshot shows the Wireshark interface with a packet list and protocol details. The packet list table is as follows:

No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
3	0.176	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
4	0.176	SNMP	192.168.5.254	161				report 1.3.6.1.6.3.15.1.1.2.0
5	0.325	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
6	0.326	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
7	0.490	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
8	0.490	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
9	0.675	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
10	0.767	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
11	0.945	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
12	0.946	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
13	1.133	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
14	1.134	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
15	1.317	SNMP	192.168.10.10	65484				encryptedPDU: privKey Unknown
16	1.318	SNMP	192.168.5.254	161				encryptedPDU: privKey Unknown
17	17.595	SNMP	192.168.10.10	62008				getBulkRequest
18	17.595	SNMP	192.168.5.254	161				report 1.3.6.1.6.3.15.1.1.4.0
19	17.749	SNMP	192.168.10.10	62008				
20	17.749	SNMP	192.168.5.254	161				
21	17.898	SNMP	192.168.10.10	62008				
22	17.899	SNMP	192.168.5.254	161	62008	192.168.10.10	070	
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	

The protocol details pane shows the following structure:

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

The 'Protocol Preferences' menu is open, and the 'Users Table...' option is highlighted. Other options include 'Show SNMP OID in info column', 'Reassemble SNMP-over-TCP messages spanning multiple TCP segments', 'Display dissected variables inside SNMP tree', 'Enterprise Specific Trap Types...', 'SNMP UDP port: 161...', 'SNMP TCP port: 161...', and 'Disable SNMP...'.

In der Tabelle "SNMP-Benutzer" wurden der SNMP-Benutzername Version 3, das Authentifizierungsmodell, das Authentifizierungskennwort, das Datenschutzprotokoll und das Datenschutzkennwort angegeben (die tatsächlichen Anmeldeinformationen werden unten nicht angezeigt):

SNMP Users

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

[C:\Users\igasimov\AppData\Roaming\Wireshark\profiles\Profile1](#)

Nachdem die SNMP-Benutzereinstellungen übernommen wurden, zeigte Wireshark entschlüsselte SNMP-PDUs an:

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	1 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	1 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	1 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	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.8.1.8
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.8.1.8
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.17.1.1 1.3.6.1.4.1.9.9.221.1.1.1.18.1.8
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.17.1.1 1.3.6.1.4.1.9.9.221.1.1.1.18.1.8
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.19.1.1 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.19.1.1 1.3.6.1.4.1.9.9.221.1.1.1.20.1.8
15	1.317	SNMP	192.168.10.10	65484	161	192.168.5.254	205	1 getBulkRequest 1.3.6.1.4.1.9.9.392.1.1.1.0 1.3.6.1.4.1.9.9.221.1.1.1.0
16	1.318	SNMP	192.168.5.254	161	65484	192.168.10.10	513	2 get-response 1.3.6.1.4.1.9.9.392.1.1.1.0 1.3.6.1.4.1.9.9.221.1.1.1.0
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	1 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	1 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	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8
23	18.094	SNMP	192.168.10.10	62008	161	192.168.5.254	205	1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
24	18.094	SNMP	192.168.5.254	161	62008	192.168.10.10	560	2 get-response 1.3.6.1.4.1.9.9.221.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.6.1.8
25	18.290	SNMP	192.168.10.10	62008	161	192.168.5.254	205	1 getBulkRequest

```

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)
  
```

Wichtigste Punkte:

1. Die SNMP-Überwachungstools verwendeten SNMP getBulkRequest, um die übergeordnete OID 1.3.6.1.4.1.9.9.221.1 und die zugehörigen OIDs abzufragen und zu durchlaufen.
2. Die FTD antwortete auf jede getBulkRequest mit get-response, die OIDs im Zusammenhang mit 1.3.6.1.4.1.9.9.221.1 enthielt.

Maßnahme 2: Identifizieren der SNMP OIDs

[SNMP Object Navigator](#) hat gezeigt, dass die OID 1.3.6.1.4.1.9.9.221.1 zur Management Information Base (MIB) mit dem Namen **CISCO-ENHANCED-MEMPOOL-MIB** gehört, wie im Bild gezeigt:

The screenshot shows the 'SNMP Object Navigator' interface. At the top, there are navigation links: HOME, SUPPORT, TOOLS & RESOURCES, and a highlighted 'SNMP Object Navigator' button. Below these are buttons for 'TRANSLATE/BROWSE', 'SEARCH', 'DOWNLOAD MIBS', and 'MIB SUPPORT - SW'. The main section is titled 'Translate | Browse The Object Tree'. It contains a text input field with the value '1.3.6.1.4.1.9.9.221.1' and a 'Translate' button. To the right, it shows an example: 'examples - OID: 1.3.6.1.4.1.9.9.27 Object Name: ifIndex'. Below this is the 'Object Information' section, which includes a table with the following data:

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

Below the table is the 'OID Tree' section, which shows a hierarchical tree structure. The current view is set to 2 levels of hierarchy above the object. The tree structure is as follows:

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

So zeigen Sie die OIDs in einem für Menschen lesbaren Format in Wireshark an:

1. Laden Sie MIB **CISCO-ENHANCED-MEMPOOL-MIB** und die dazugehörigen Abhängigkeiten herunter, wie im Bild gezeigt:

SNMP Object Navigator

[HOME](#)

[SUPPORT](#)

[TOOLS & RESOURCES](#)

SNMP Object Navigator

TRANSLATE/BROWSE

SEARCH

DOWNLOAD MIBS

MIB SUPPORT - SW

View MIB dependencies and download MIB or view MIB contents

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

CISCO-ENHANCED-MEMPOOL-MIB

List matching MIBs

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

Step 2: Select a function:

- View MIB dependencies and download MIB
- View MIB contents

Submit

SNMP Object Navigator

[HOME](#)[SUPPORT](#)[TOOLS & RESOURCES](#)**SNMP Object Navigator**[TRANSLATE/BROWSE](#)[SEARCH](#)[DOWNLOAD MIBS](#)[MIB SUPPORT - SW](#)**CISCO-ENHANCED-MEMPOOL-MIB**

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

Compile the MIB

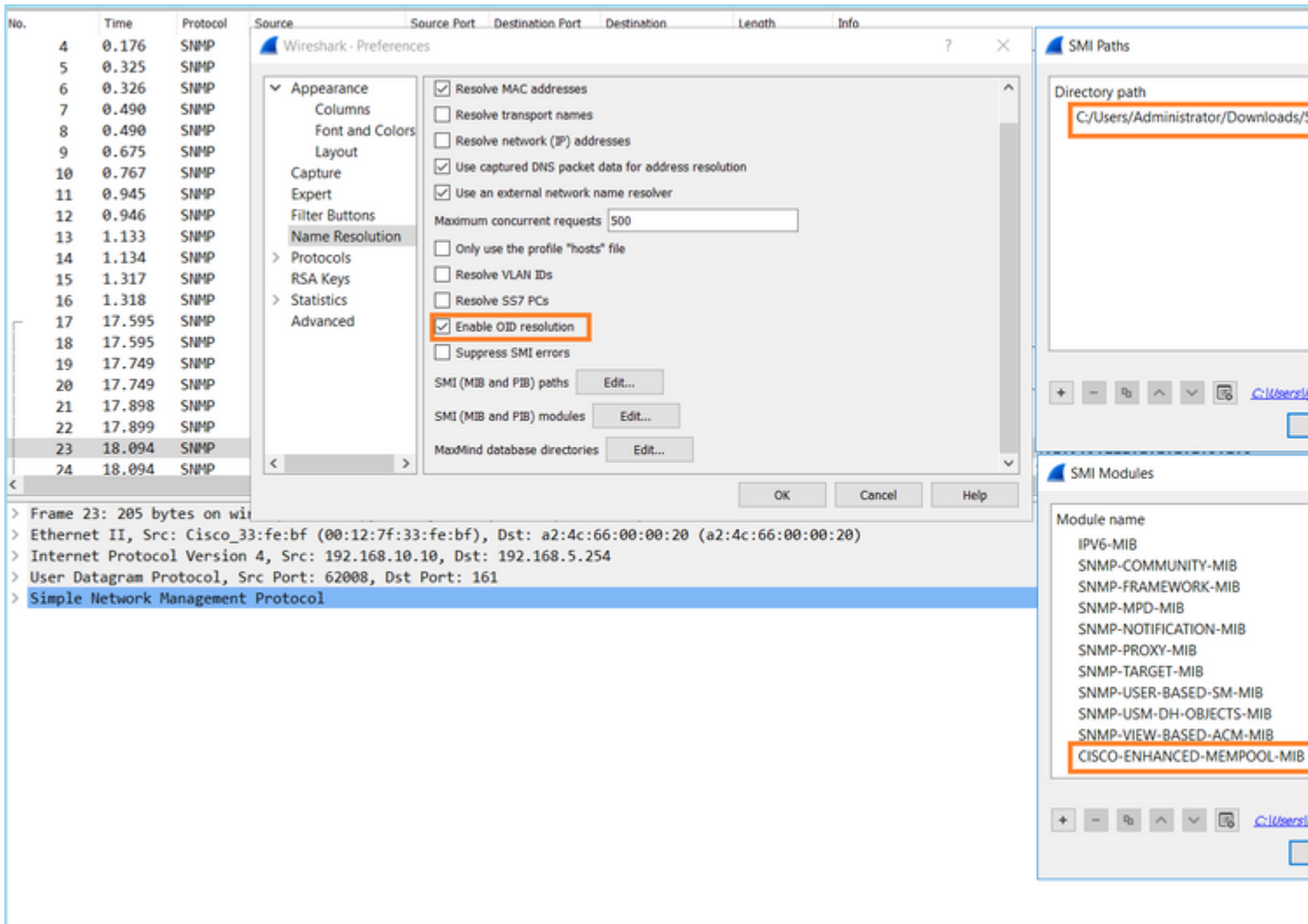
Before you can compile CISCO-ENHANCED-MEMPOOL-MIB, you need to compile the MIBs listed below in the order listed.

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

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

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

2. In Wireshark im Fenster **Bearbeiten** > **Voreinstellungen** > **Namensauflösung** ist **OID-Auflösung aktivieren** aktiviert. Geben Sie im Fenster **SMI (MIB- und PIB-Pfade)** den Ordner mit den heruntergeladenen MIBs und in **SMI (MIB- und PIB-Module)** an. Die CISCO-ENHANCED-MEMPOOL-MIB wird automatisch zur Modulliste hinzugefügt:



3. Nach dem Neustart von Wireshark wird die OID-Auflösung aktiviert:

No.	Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report SNMP-USER-BASED-SM-MIB::usmStatsUnknownEngineID
3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMIBObjec
4	0.176	SNMP	192.168.5.254	161	65484	192.168.10.10	192	report SNMP-USER-BASED-SM-MIB::usmStatsNotInTimeWindow
5	0.325	SNMP	192.168.10.10	65484	161	192.168.5.254	199	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMIBObjec
6	0.326	SNMP	192.168.5.254	161	65484	192.168.10.10	678	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolTyp
7	0.490	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolAl
8	0.490	SNMP	192.168.5.254	161	65484	192.168.10.10	560	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolAl
9	0.675	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPool
10	0.767	SNMP	192.168.5.254	161	65484	192.168.10.10	610	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUs
11	0.945	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPool
12	0.946	SNMP	192.168.5.254	161	65484	192.168.10.10	584	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolUs
13	1.133	SNMP	192.168.10.10	65484	161	192.168.5.254	205	getBulkRequest CISCO-ENHANCED-MEMPOOL-MIB::cempMemPool
14	1.134	SNMP	192.168.5.254	161	65484	192.168.10.10	588	get-response CISCO-ENHANCED-MEMPOOL-MIB::cempMemPoolEn

```

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

```

Basierend auf der entschlüsselten Ausgabe der Erfassungsdatei wurden vom SNMP-Überwachungstool in regelmäßigen Abständen (10-Sekunden-Intervall) Daten zur Nutzung von Speicherpools auf dem FTD abgefragt. Wie im TechNote-Artikel [ASA SNMP Polling for Memory-Related Statistics](#) erläutert, führt das Polling der Nutzung des globalen gemeinsamen Pools (GSP) mit SNMP zu einer hohen CPU-Auslastung. In diesem Fall wurde aus den Erfassungen deutlich, dass die Nutzung des globalen gemeinsamen Pools regelmäßig als Teil von SNMP getBulkRequest primitive abgefragt wurde.

Um die durch den SNMP-Prozess verursachten CPU-Hogs zu minimieren, wurde empfohlen, die im Artikel genannten Schritte zur Risikominimierung für CPU-Hogs für SNMP zu befolgen und ein Abfragen der OIDs in Bezug auf GSP zu vermeiden. Ohne die SNMP-Abfrage für die OIDs, die sich auf GSP beziehen, wurden keine CPU-Hogs beobachtet, die durch den SNMP-Prozess verursacht wurden, und die Rate der Überläufe hat sich deutlich verringert.

Zugehörige Informationen

- [Cisco FirePOWER Management Center - Konfigurationsleitfäden](#)
- [Klären der Regelaktionen der Firepower Threat Defense-Richtlinien zur Zugriffskontrolle](#)
- [Arbeiten mit Firepower Threat Defense Captures und Packet Tracer](#)
- [Wireshark lernen](#)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.