Analyser les captures du pare-feu Firepower pour résoudre les problèmes réseau

Table des matières

Introduction
Conditions préalables
Exigences
Composants utilisés
Informations générales
Comment collecter et exporter des cantures sur la gamme de produite NGEW 2
Collecter les captures FXOS
Activer et collecter les captures Lina FTD
Activer et collecter les captures FTD Snort
<u>Dépannage</u>
Cas 1 . Pas de SYN TCP sur l'interface de sortie
Analyse de capture
Actions recommandées
Synthèse des causes possibles et des actions recommandées
Cas 2 . TCP SYN du client, TCP RST du serveur
Analyse de capture
Actions recommandées
Cas 3 . Connexion TCP en trois étapes + RST à partir d'un terminal
Analyse de capture
3.1 - Connexion TCP en trois étapes + RST différé du client
Actions recommandées
3.2 - Connexion TCP en trois étapes + FIN/ACK retardé du client + RST retardé du serveur
Actions recommandées
<u>3.3 - Connexion TCP en trois étapes + RST différé du client</u>
Actions recommandees
<u>3.4 - Connexion TCP en trois étapes + RST immédiat à partir du serveur</u>
Cas 4 . TCP RST a partir du client
Analyse de capture
Cas 5. Transfert TCP lent (scenario 1)
Scénario 2. Transfert rapide
Cas 6 Transfort TCP lont (scénario 2)
Actions recommandées
Exportez la canture pour vérifier la différence de temps entre les paquets d'entrée et de
sortieCas 7. Problème de connectivité TCP (corruption de paquet)

Analyse de capture

Actions recommandées
Cas 8 . Problème de connectivité UDP (paquets manquants)
Analyse de capture
Actions recommandées
Analyse de capture
Actions recommandées
Cas 10. Problème de connectivité HTTPS (scénario 2)
Analyse de capture
Actions recommandées
Analyse de capture
Actions recommandées
Cas 12. Problème de connectivité intermittent (empoisonnement ARP)
Analyse de capture
Actions recommandées
Cas 13 . Identifier les identificateurs d'objet SNMP (OID) qui provoquent des erreurs de CPU
Analyse de capture
Actions recommandées
Informations connexes

Introduction

Ce document décrit diverses techniques d'analyse de capture de paquets qui visent à dépanner efficacement les problèmes de réseau.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Architecture de plate-forme Firepower
- Journaux NGFW
- traceur de paquets de pare-feu de nouvelle génération

En outre, avant de commencer à analyser les captures de paquets, il est vivement conseillé de respecter les exigences suivantes :

- Connaître le fonctionnement du protocole Ne commencez pas à vérifier une capture de paquets si vous ne comprenez pas le fonctionnement du protocole capturé.
- Connaître la topologie Vous devez connaître les périphériques de transit de bout en bout. Si cela n'est pas possible, vous devez au moins connaître les périphériques en amont et en aval.
- Connaître l'appliance : vous devez savoir comment votre périphérique gère les paquets, quelles sont les interfaces impliquées (entrée/sortie), quelle est l'architecture du périphérique et quels sont les différents points de capture.
- Connaître la configuration Vous devez savoir comment un flux de paquets est censé être géré par le périphérique en termes de :
 - Interface de routage/sortie

- Stratégies appliquées
- Traduction d'adresses réseau (NAT)
- Connaître les outils disponibles En plus des captures, il est recommandé d'être prêt à appliquer d'autres outils et techniques (comme la journalisation et les traceurs) et, si nécessaire, de les corréler avec les paquets capturés.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- La plupart des scénarios sont basés sur FP4140 exécutant le logiciel FTD 6.5.x.
- FMC exécutant le logiciel 6.5.x.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

La capture de paquets est l'un des outils de dépannage les plus négligés actuellement disponibles. Le TAC Cisco résout quotidiennement de nombreux problèmes liés à l'analyse des données capturées.

L'objectif de ce document est d'aider les ingénieurs réseau et de sécurité à identifier et à dépanner les problèmes réseau courants, principalement en se basant sur l'analyse de capture de paquets.

Tous les scénarios présentés dans ce document sont basés sur des cas réels d'utilisateurs observés dans le centre d'assistance technique de Cisco (TAC).

Le document couvre les captures de paquets du point de vue du pare-feu de nouvelle génération Cisco (NGFW), mais les mêmes concepts s'appliquent également à d'autres types de périphériques.

Comment collecter et exporter des captures sur la gamme de produits NGFW ?

Dans le cas d'un appareil Firepower (1xxx, 21xx, 41xx, 93xx) et d'une application Firepower Threat Defense (FTD), un traitement de paquets peut être visualisé comme illustré dans l'image.



- 1. Un paquet entre dans l'interface d'entrée et est géré par le commutateur interne du châssis.
- 2. Le paquet entre dans le moteur Lina FTD qui effectue principalement des vérifications L3/L4.
- 3. Si la politique exige que le paquet soit inspecté par le moteur Snort (principalement inspection L7).
- 4. Le moteur Snort renvoie un verdict pour le paquet.
- 5. Le moteur LINA abandonne ou transfère le paquet en fonction du verdict du renifleur.
- 6. Le paquet sort du châssis par le commutateur interne du châssis.

Sur la base de l'architecture illustrée, les captures FTD peuvent être effectuées à trois (3) endroits différents :

- FXOS
- moteur FTD Lina
- Moteur FTD Snort

Collecter les captures FXOS

Le processus est décrit dans ce document :

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

Les captures FXOS ne peuvent être prises que dans la direction d'entrée à partir du point de vue du commutateur interne sont montrées dans l'image ici.



Dans le cas présent, il s'agit de deux points de capture par direction (en raison de l'architecture

interne du commutateur).



Les paquets capturés aux points 2, 3 et 4 ont une étiquette de réseau virtuel (VNTag).

Remarque : les captures au niveau du châssis FXOS sont uniquement disponibles sur les plates-formes FP41xx et FP93xx. Les modèles FP1xxx et FP21xx n'offrent pas cette fonctionnalité.

Activer et collecter les captures Lina FTD

Principaux points de capture :

- Interface d'entrée
- Interface de sortie
- Chemin de sécurité accéléré (ASP)



Vous pouvez utiliser l'interface utilisateur FMC (Firepower Management Center User Interface) ou l'interface de ligne de commande FTD pour activer et collecter les captures FTD Lina.

Activez la capture à partir de l'interface CLI sur l'interface INSIDE :

<#root>

firepower#

capture CAPI interface INSIDE match icmp host 192.168.103.1 host 192.168.101.1

Cette capture correspond au trafic entre les adresses IP 192.168.103.1 et 192.168.101.1 dans les deux directions.

Activez la capture ASP pour voir tous les paquets abandonnés par le moteur FTD Lina :

<#root>

firepower#

```
capture ASP type asp-drop all
```

Exporter une capture FTD Lina vers un serveur FTP :

<#root>

firepower#

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

Exportez une capture FTD Lina vers un serveur TFTP :

<#root>

firepower#

copy /pcap capture:CAPI tftp://192.168.78.73

À partir de la version FMC 6.2.x, vous pouvez activer et collecter les captures FTD Lina à partir de l'interface utilisateur FMC.

Voici une autre façon de collecter des captures FTD à partir d'un pare-feu géré par FMC.

Étape 1

Dans le cas d'une capture LINA ou ASP, copiez la capture sur le disque FTD.

<#root>

firepower#

copy /pcap capture:capin disk0:capin.pcap

Source capture name [capin]?

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

Étape 2

Accédez au mode expert, localisez la capture enregistrée et copiez-la dans /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

Étape 3

Connectez-vous au FMC qui gère le FTD et accédez à Périphériques > Gestion des périphériques. Localisez le périphérique FTD et sélectionnez l'icône Troubleshoot :



cisco	Firepower Management Center System / Health / Health Monitor Appliance	۹	С	Verview	Analysis	Policies
Health	Monitor					
	Appliance					
9	mzafeiro_FP2110-2	G	enerat	te Troublesho	oting Files	
		A	dvanc	ed Troublesh	ooting	

Spécifiez le nom du fichier de capture et sélectionnez Télécharger :

Firepower Management Center System / Health / AT File Download	Q	Overview	Analysis	Policies	Devices	Objects	AMP	Intelligence
Advanced Troubleshooting mzafeiro_FP2110-2 File Download Threat Defense CLI Packet Tra	icer	Capture w/Trac	е					
			File	apin.pcap			Bad	ck Download

Pour plus d'exemples sur la façon d'activer/collecter des captures à partir de l'interface utilisateur FMC, consultez ce document :

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

Activer et collecter les captures FTD Snort

Le point de capture est illustré dans l'image ci-contre.



Activer la capture de niveau Snort :

```
<#root>
>
capture-traffic

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

Please specify tcpdump options desired.
(or enter '?' for a list of supported options)
Options:
-n host 192.168.101.1
```

Pour écrire la capture dans un fichier nommé capture.pcap et la copier via FTP sur un serveur distant :

<#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 >

Pour plus d'exemples de capture de niveau Snort qui incluent différents filtres de capture, consultez ce document :

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

Dépannage

Cas 1 . Pas de SYN TCP sur l'interface de sortie

La topologie est illustrée dans l'image ci-dessous :



Description du problème : HTTP ne fonctionne pas

Flux affecté :

Adresse IP source : 192.168.0.100

Adresse IP de destination : 10.10.1.100

Protocole : TCP 80

Analyse de capture

Activez les captures sur le moteur FTD LINA :

<#root>

firepower#

capture CAPI int INSIDE match ip host 192.168.0.100 host 10.10.1.100

firepower#



Captures - Scénario fonctionnel :

En tant que base, il est toujours très utile de disposer de captures à partir d'un scénario fonctionnel.

Capture prise sur l'interface INSIDE du pare-feu de nouvelle génération, comme illustré dans l'image :

	CAPI-working.pcap							
E	ile <u>E</u> dit <u>V</u> iew <u>G</u>	o <u>Capture</u> Analyze	Statistics Telephony	Wireless To	ols <u>H</u> elp			
1		🖹 🕅 🙆 🖉 🛶 🖬		0001				
4	tcp.stream eq 1							
Ν	o. Time	Source	Destination	Protocol L	ength Info			
r	2 0.250878	192.168.0.100	10.10.1.100	ТСР	66 1779 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1			
	3 0.001221	10.10.1.100	192.168.0.100	TCP	66 80 → 1779 [SYN, ACK] seq=0 Ack=1 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1			
	4 0.000488	192.168.0.100	10.10.1.100	TCP	54 1779 → 80 [ACK] Seq=1 Ack=1 Win=66240 Len=0			
	5 0.000290	192.168.0.100	10.10.1.100	HTTP	369 GET / HTTP/1.1			
	6 0.002182	10.10.1.100	192.168.0.100	HTTP	966 HTTP/1.1 200 OK (text/html)			
	7 0.066830	192.168.0.100	10.10.1.100	HTTP	331 GET /welcome.png HTTP/1.1			
	8 0.021727	10.10.1.100	192.168.0.100	TCP	1434 80 → 1779 [ACK] Seq=913 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]			
	9 0.000000	10.10.1.100	192.168.0.100	TCP	1434 80 → 1779 [ACK] Seq=2293 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]			
	10 0.000626	192.168.0.100	10.10.1.100	TCP	54 1779 → 80 [ACK] Seq=593 Ack=3673 Win=66240 Len=0			
5	Ename 2: 66 hutes on wine (528 hits) 66 hutes centured (528 hits)							
5	Friende II. Social Cisco ferfords (S. de 35: ferfords) Det: Cisco f6:14:ae (00:he:75: f6:14:ae)							
5	Internet Prot	ocol Version 4. S	rc: 192.168.0.100	Dst: 10.1	0.1.100			
5	Transmission	Control Protocol.	Src Port: 1779, D	st Port: 8	0. Seq: 0. Len: 0			
ľ			5. c . c. c. 1775, 5	se torer o	-,, -, -,			

Principaux points :

- 1. Connexion TCP en trois étapes.
- 2. Échange de données bidirectionnel.
- 3. Aucun délai entre les paquets (en fonction de la différence de temps entre les paquets).
- 4. L'adresse MAC source est le périphérique en aval correct.

Capture prise sur l'interface NGFW OUTSIDE, est montré dans l'image ici :

4	CAPO-working.pc	ap						
<u>F</u> ile	Edit View Q	o <u>C</u> apture <u>A</u> nalyze	Statistics Telephony	<u>W</u> ireless	<u>I</u> ools <u>H</u> elp			
	🔳 🖉 🕥 📕	🗎 🖹 🎑 🔍 🖛 =	• 🖻 Ŧ 🛓 📃 📃		. II			
🔳 te	cp.stream eq 1							
No.	Time	Source	Destination	Protocol	Length Info			
Г	2 0.250787	192.168.0.100	10.10.1.100	TCP	70 1779 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1			
	3 0.000534	10.10.1.100	192.168.0.100	TCP	70 80 → 1779 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1			
	4 0.000564	192.168.0.100	10.10.1.100	TCP	58 1779 → 80 [ACK] Seq=1 Ack=1 Win=66240 Len=0			
	5 0.000534	192.168.0.100	10.10.1.100	HTTP	373 GET / HTTP/1.1			
	6 0.001663	10.10.1.100	192.168.0.100	HTTP	970 HTTP/1.1 200 OK (text/html)			
	7 0.067273	192.168.0.100	10.10.1.100	HTTP	335 GET /welcome.png HTTP/1.1			
	8 0.021422	10.10.1.100	192.168.0.100	TCP	1438 80 → 1779 [ACK] Seq=913 Ack=593 Win=65792 Len=1380 [TCP segment df a reassembled PDU]			
	9 0.000015	10.10.1.100	192.168.0.100	TCP	1438 80 → 1779 [ACK] Seq=2293 Ack=593 Win=65792 Len=1380 [TCP segment of a reassembled PDU]			
<								
>	Frame 2: 70 b	oytes on wire (560	bits), 70 bytes c	aptured	(560 bits)			
>	Ethernet II,	Src: Cisco f6:1d:	8e (00:be:75:f6:1d	:8e), Ds	t: Cisco fc:fc:d8 (4c:4e:35:fc:fc:d8)			
>	> 802.10 Virtual LAN, PRI: 0, DEI: 0, ID: 202							
>	Internet Prot	cocol Version 4, S	rc: 192.168.0.100,	Dst: 10	.10.1.100			
>	Transmission	Control Protocol,	Src Port: 1779, D	st Port:	80, Seq: 0, Len: 0			

Principaux points :

- 1. Mêmes données que dans la capture CAPI.
- 2. L'adresse MAC de destination est le périphérique en amont correct.

Captures - Scénario non fonctionnel

À partir de l'interface de ligne de commande du périphérique, les captures ressemblent à ceci :

<#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

Contenu CAPI :

<#root>

firepower#

show capture CAPI

6 packets captured

1: 11:47:46.911482 192.168.0.100.3171 > 10.10.1.100.80:

s

```
1089825363:1089825363(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
2: 11:47:47.161902 192.168.0.100.3172 > 10.10.1.100.80:
s
3981048763:3981048763(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
3: 11:47:49.907683 192.168.0.100.3171 > 10.10.1.100.80:
s
1089825363:1089825363(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
4: 11:47:50.162757 192.168.0.100.3172 > 10.10.1.100.80:
s
3981048763:3981048763(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
5: 11:47:55.914640 192.168.0.100.3171 > 10.10.1.100.80:
```

s

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

s

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

<#root>

firepower#

show capture CAPO

0 packet captured

0 packet shown

Voici l'image de la capture CAPI dans Wireshark :

No.		Time	Source	Destination	Protocol	Length	Info
Г	1	0.000000	192.168.0.100	10.10.1.100	ТСР	66	3171 → 80 [SYN] Seq=0 Win=8192 Len=0=1460 WS=4 SACK_PERM=1
	2	0.250420	192.168.0.100	10.10.1.100	тср	66	3172 → 80 [SYN] Seq= <u>0 Win=</u> 8192 Len=05=1460 WS=4 SACK_PERM=1
	3	2.745781	192.168.0.100	10.10.1.100	тср	66	[TCP Retransmission] 3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
		0.255074	192.168.0.100	10.10.1.100	тср	66	[TCP Retransmission] 3172 > 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
L		5.751883	192.168.0.100	10.10.1.100	тср	62	[TCP Retransmissi3171 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
	6	0.250070	192.168.0.100	10.10.1.100	тср	62	[TCP Retransmissi → 3172 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
	7	3					
>	Fram	- 1: 66 b	vtes on wire (528	bits), 66 bytes ca	ptured ((528 bits	5)
>	Ethernet II Acc: Cisco fc:fc:d8 (4c:4e:35:fc:fc:d8) Dst: Cisco f6:1d:ae (00:be:75:f6:1d:ae)						
>	Internet Pr 🔁 col 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						

Principaux points :

- 1. Seuls les paquets TCP SYN sont visibles (pas de connexion TCP en trois étapes).
- Il est impossible d'établir 2 sessions TCP (ports source 3171 et 3172). Le client source renvoie les paquets TCP SYN. Ces paquets retransmis sont identifiés par Wireshark comme des retransmissions TCP.

- 3. Les retransmissions TCP ont lieu toutes les ~3 puis 6 secondes, etc.
- 4. L'adresse MAC source provient du périphérique en aval correct.

Sur la base des deux captures, on peut conclure que :

- Un paquet d'un 5-tuple spécifique (IP src/dst, port src/dst, protocole) arrive sur le pare-feu sur l'interface attendue (INSIDE).
- Un paquet ne quitte pas le pare-feu sur l'interface attendue (OUTSIDE).

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Vérifiez la trace d'un paquet émulé.

Utilisez l'outil packet-tracer pour voir comment un paquet est censé être traité par le pare-feu. Si le paquet est abandonné par la politique d'accès du pare-feu, la trace du paquet émulé ressemble à ce résultat :

<#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 Confia: 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-status: up output-line-status: up Output-line-status: up Drop-reason: (acl-drop) Flow is denied by configured rule, Drop-location: frame 0x00005647a4f4b120 flow

Action 2. Vérifiez les traces des paquets actifs.

Activez le suivi des paquets pour vérifier comment les paquets TCP SYN réels sont traités par le pare-feu. Par défaut, seuls les 50 premiers paquets entrants sont suivis :

<#root>

firepower#

capture CAPI trace

Effacez la mémoire tampon de capture :

<#root>

firepower#

clear capture /all

Si le paquet est abandonné par la politique d'accès du pare-feu, la trace ressemble à ce résultat :

 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

Action 3. Vérifiez les journaux FTD Lina.

Pour configurer Syslog sur FTD via FMC, consultez ce document :

https://www.cisco.com/c/en/us/support/docs/security/firepower-ngfw/200479-Configure-Loggingon-FTD-via-FMC.html Il est fortement recommandé de configurer un serveur Syslog externe pour les journaux FTD Lina. Si aucun serveur Syslog distant n'est configuré, activez les journaux de mémoire tampon locale sur le pare-feu pendant le dépannage. La configuration du journal présentée dans cet exemple est un bon point de départ :

<#root>

firepower#

show run logging

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

Réglez le pager terminal sur 24 lignes afin de contrôler le pager terminal :

<#root>

firepower#

terminal pager 24

Effacez la mémoire tampon de capture :

<#root>

firepower#

clear logging buffer

Testez la connexion et vérifiez les journaux avec un filtre d'analyse. Dans cet exemple, les paquets sont abandonnés par la politique d'accès du pare-feu :

<#root>

firepower#

show logging | include 10.10.1.100

Oct 09 2019 12:55:51: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3696 dst OUTSIDE:10.10.1.100/80 Oct 09 2019 12:55:51: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3697 dst OUTSIDE:10.10.1.100/80 Oct 09 2019 12:55:54: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3696 dst OUTSIDE:10.10.1.100/80 Oct 09 2019 12:55:54: %FTD-4-106023: Deny tcp src INSIDE:192.168.0.100/3697 dst OUTSIDE:10.10.1.100/80 Mesure 4. Vérifiez que le pare-feu ASP abandonne.

Si vous suspectez que le paquet est abandonné par le pare-feu, vous pouvez voir les compteurs de tous les paquets abandonnés par le pare-feu au niveau logiciel :

<#root>	
firepower#	
show asp drop	
Frame drop: No route to host (no-route) Flow is denied by configured rule (acl-drop)	234 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	

Vous pouvez activer les captures pour afficher toutes les pertes de niveau logiciel ASP :

<#root>

firepower#

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

Conseil : si vous n'êtes pas intéressé par le contenu du paquet, vous pouvez capturer uniquement les en-têtes de paquet (option en-têtes uniquement). Cela vous permet de capturer beaucoup plus de paquets dans la mémoire tampon de capture. En outre, vous pouvez augmenter la taille de la mémoire tampon de capture (par défaut, elle est de 500 Ko) jusqu'à une valeur de 32 Mo (option de mémoire tampon). Enfin, à partir de la version FTD 6.3, l'option file-size vous permet de configurer un fichier de capture jusqu'à 10 Go. Dans ce cas, vous ne pouvez voir le contenu de la capture qu'au format pcap.

Pour vérifier le contenu de la capture, vous pouvez utiliser un filtre pour affiner votre recherche :

<#root>						
firepower#						
show capture ASP inclu	ude 10.10.1.100					
18: 07:51:57.823672 19: 07:51:58.074291	192.168.0.100.12410 > 192.168.0.100.12411 >	10.10.1.100.80: 10.10.1.100.80:	S 1870382552:2 S 2006489005:2	1870382552(0) 2006489005(0)	win 8192 win 8192	<mss <mss< th=""></mss<></mss

26:	07:52:00.830370	192.168.0.100.12410	>	10.10.1.100.80:	S	1870382552:1870382552(0)	win	8192	<mss< th=""></mss<>
29:	07:52:01.080394	192.168.0.100.12411	>	10.10.1.100.80:	S	2006489005:2006489005(0)	win	8192	<mss< td=""></mss<>
45:	07:52:06.824282	192.168.0.100.12410	>	10.10.1.100.80:	S	1870382552:1870382552(0)	win	8192	<mss< td=""></mss<>
46:	07:52:07.074230	192.168.0.100.12411	>	10.10.1.100.80:	S	2006489005:2006489005(0)	win	8192	<mss< td=""></mss<>

Dans ce cas, puisque les paquets sont déjà tracés au niveau de l'interface, la raison de l'abandon n'est pas mentionnée dans la capture ASP. N'oubliez pas qu'un paquet ne peut être suivi qu'à un seul endroit (interface d'entrée ou abandon ASP). Dans ce cas, il est recommandé de prendre plusieurs abandons ASP et de définir une raison d'abandon ASP spécifique. Voici une approche recommandée :

1. Effacez les compteurs d'abandon ASP actuels :

<#root> firepower# clear asp drop

2. Envoyez le flux que vous dépannez via le pare-feu (exécutez un test).

3. Vérifiez à nouveau les compteurs de dépôt ASP et notez ceux qui ont augmenté.

<#root> firepower# show asp drop Frame drop: No route to host (no-route) 234 Flow is denied by configured rule (acl-drop) 71

4. Activez la ou les captures ASP pour les abandons spécifiques affichés :

<#root>

firepower#

capture ASP_NO_ROUTE type asp-drop no-route

firepower#

5. Envoyez le flux que vous dépannez via le pare-feu (exécutez un test).

6. Vérifiez les captures ASP. Dans ce cas, les paquets ont été abandonnés en raison d'une route absente :

<#root>

firepower#

show capture ASP_NO_ROUTE | include 192.168.0.100.*10.10.1.100

```
93: 07:53:52.381663192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss</td>95: 07:53:52.632337192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss</td>101: 07:53:55.375392192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss</td>102: 07:53:55.626386192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss</td>116: 07:54:01.376231192.168.0.100.12417 > 10.10.1.100.80: S 3451917925:3451917925(0) win 8192 <mss</td>117: 07:54:01.626310192.168.0.100.12418 > 10.10.1.100.80: S 1691844448:1691844448(0) win 8192 <mss</td>
```

Action 5. Vérifiez la table de connexion FTD Lina.

Il peut y avoir des cas où vous vous attendez à ce que le paquet sorte de l'interface 'X', mais pour quelque raison que ce soit, il sort de l'interface 'Y'. La détermination de l'interface de sortie du pare-feu est basée sur cet ordre de fonctionnement :

- 1. Recherche de connexion établie
- Recherche NAT (Network Address Translation) : la phase UN-NAT (destination NAT) est prioritaire sur la recherche PBR et la recherche de route.
- 3. Routage basé sur des politiques (PBR)
- 4. Recherche dans la table de routage

Pour vérifier la table de connexion FTD :

```
<#root>
firepower#
show conn
2 in use, 4 most used
Inspect Snort:
    preserve-connection: 2 enabled, 0 in effect, 4 most enabled, 0 most in effect
TCP
DMZ
10.10.1.100:
80
```

```
INSIDE

192.168.0.100:

11694

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

aA N1

TCP

DMZ

10.10.1.100:80

INSIDE

192.168.0.100:

11693

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

aA N1
```

Principaux points :

- Selon les indicateurs (Aa), la connexion est embryonnaire (semi-ouverte seul TCP SYN a été vu par le pare-feu).
- En fonction des ports source/de destination, l'interface d'entrée est INSIDE et l'interface de sortie est DMZ.

Ceci peut être visualisé dans l'image ici :



Remarque : comme toutes les interfaces FTD ont un niveau de sécurité de 0, l'ordre des interfaces dans la sortie de show conn est basé sur le numéro d'interface. Plus précisément, l'interface avec le numéro vpif-num supérieur (numéro d'interface de plate-forme virtuelle) est sélectionnée comme interne, tandis que l'interface avec le numéro vpif-num inférieur est sélectionnée comme externe. Vous pouvez voir la valeur vpif de l'interface avec la

commande show interface detail. Amélioration connexe, ID de bogue Cisco <u>CSCvi15290</u> ENH : FTD affiche la directionnalité de connexion dans la sortie « show conn » de FTD

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
```

Remarque : à partir de la version 6.5 du logiciel Firepower, ASA version 9.13.x, les résultats des commandes show conn long et show conn detail fournissent des informations sur l'initiateur et le répondeur de la connexion

Résultat 1 :

<#root>

firepower#

show conn long

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

Initiator: 192.168.1.100, Responder: 192.168.2.200

Connection lookup keyid: 228982375

Résultat 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

En outre, la commande show conn long affiche les IP NATed entre parenthèses dans le cas d'une traduction d'adresses réseau :

<#root>

firepower#

show conn long

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

Mesure no 6. Vérifiez le cache ARP (Address Resolution Protocol) du pare-feu.

Si le pare-feu ne peut pas résoudre le saut suivant, il abandonne silencieusement le paquet d'origine (TCP SYN dans ce cas) et envoie continuellement des requêtes ARP jusqu'à ce qu'il résolve le saut suivant.

Afin de voir le cache ARP du pare-feu, utilisez la commande :

<#root>

firepower#

show arp

En outre, pour vérifier s'il existe des hôtes non résolus, vous pouvez utiliser la commande suivante :

<#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</pre>
```

Unresolved hosts:

1

```
< this is the current status
```

Maximum Unresolved hosts: 2

Si vous souhaitez vérifier davantage l'opération ARP, vous pouvez activer une capture spécifique à ARP :

<#root>						
firepower#						
capture ARP ethernet-type arp interface OUTSIDE						
firepower#						
show capture ARP						
4: 07:15:16.877914	802.1Q vlan#202 P0 arp					
who-has 192.168.2.72 tell 19	2.168.2.50					
5: 07:15:18.020033	802.1Q vlan#202 PO arp who-has 192.168.2.72 tell 192.168.2.50					

Dans ce résultat, le pare-feu (192.168.2.50) tente de résoudre le tronçon suivant (192.168.2.72), mais il n'y a pas de réponse ARP



Le résultat ci-dessous montre un scénario fonctionnel avec une résolution ARP appropriée :

<#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

Si aucune entrée ARP n'est en place, la trace d'un paquet SYN TCP actif indique :

<#root> firepower# show capture CAPI packet-number 1 trace 6 packets captured 1: 07:03:43.270585 192.168.0.100.11997 > 10.10.1.100.80 : S 4023707145:4023707145(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK> Phase: 1 Type: CAPTURE Subtype: Result: ALLOW Config: Additional Information: MAC Access list Phase: 2 Type: ACCESS-LIST Subtype: Result: ALLOW Config: Implicit Rule Additional Information: MAC Access list Phase: 3 Type: ROUTE-LOOKUP Subtype: Resolve Egress Interface Result: ALLOW Config: Additional Information: found next-hop 192.168.2.72 using egress ifc OUTSIDE ...

Phase: 14 Type: FLOW-CREATION Subtype: Result: ALLOW Config: Additional Information: New flow created with id 4814, packet dispatched to next module Phase: 17 Type: ROUTE-LOOKUP Subtype: Resolve Egress Interface Result: ALLOW Config: Additional Information: found next-hop 192.168.2.72 using egress ifc OUTSIDE Result: input-interface: INSIDE input-status: up input-line-status: up output-interface: OUTSIDE output-status: up output-line-status: up Action: allow

Comme vous pouvez le voir dans le résultat, la trace montre Action : allow même lorsque le saut suivant n'est pas accessible et que le paquet est silencieusement abandonné par le pare-feu ! Dans ce cas, l'outil Packet Tracer doit également être vérifié car il fournit une sortie plus précise :

<#root> firepower# packet-tracer input INSIDE tcp 192.168.0.100 1111 10.10.1.100 80 Phase: 1 Type: CAPTURE Subtype: Result: ALLOW Config: Additional Information: MAC Access list Phase: 2 Type: ACCESS-LIST Subtype: Result: ALLOW Config: Implicit Rule Additional Information: MAC Access list Phase: 3 Type: ROUTE-LOOKUP Subtype: Resolve Egress Interface Result: ALLOW

Config: Additional Information: found next-hop 192.168.2.72 using egress ifc OUTSIDE Phase: 14 Type: FLOW-CREATION Subtype: Result: ALLOW Config: Additional Information: New flow created with id 4816, packet dispatched to next module Phase: 17 Type: ROUTE-LOOKUP Subtype: Resolve Egress Interface Result: ALLOW Config: Additional Information: found next-hop 192.168.2.72 using egress ifc OUTSIDE Result: input-interface: INSIDE input-status: up input-line-status: up output-interface: OUTSIDE output-status: up output-line-status: up Action: drop Drop-reason: (no-v4-adjacency) No valid V4 adjacency, Drop-location: frame 0x00005647a4e86109 flow (NA),

Dans les versions récentes d'ASA/Firepower, le message précédent a été optimisé pour :

<#root>

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

Check ARP table (show arp) has entry for nexthop

., Drop-location: f

Synthèse des causes possibles et des actions recommandées

Si vous ne voyez qu'un paquet TCP SYN sur les interfaces d'entrée, mais qu'aucun paquet TCP SYN n'est envoyé à partir de l'interface de sortie attendue, certaines causes possibles sont :

Cause possible	Actions recommandées
Le paquet est abandonné par la politique	 Utilisez packet-tracer ou capture w/trace pour
d'accès du pare-feu.	voir comment le pare-feu gère le paquet.

	 Vérifiez les journaux du pare-feu. Vérifiez les abandons ASP du pare-feu (show asp drop ou capture type asp-drop). Vérifiez les événements de connexion FMC. Cela suppose que la journalisation est activée pour la règle. 			
Le filtre de capture est incorrect.	 Utilisez packet-tracer ou capture w/trace pour voir s'il y a une traduction NAT qui modifie l'IP source ou de destination. Dans ce cas, réglez votre filtre de capture. La sortie de la commande show conn long affiche les adresses IP NATed. 			
Le paquet est envoyé à une autre interface de sortie.	 Utilisez packet-tracer ou capture w/trace pour voir comment le pare-feu gère le paquet. Souvenez-vous de l'ordre des opérations concernant la détermination de l'interface de sortie, la connexion actuelle, l'UN-NAT, le PBR et la recherche dans la table de routage. Vérifiez les journaux du pare-feu. Vérifiez la table de connexion du pare-feu (show conn). 			
	utilisez la commande clear conn address et spécifiez le 5-tuple de la connexion que vous voulez effacer.			
Il n'y a pas de route vers la destination.	 Utilisez packet-tracer ou capture w/trace pour voir comment le pare-feu gère le paquet. Vérifiez les abandons ASP du pare-feu (show asp drop) pour la raison de non-abandon de route. 			
II n'y a aucune entrée ARP sur l'interface de sortie.	 Vérifiez le cache ARP du pare-feu (show arp). Utilisez packet-tracer pour voir s'il y a une contiguïté valide. 			
L'interface de sortie est désactivée.	Vérifiez le résultat de la commande show interface ip brief sur le pare-feu et vérifiez l'état de l'interface.			

Cas 2 . TCP SYN du client, TCP RST du serveur

Cette image présente la topologie :



Description du problème : HTTP ne fonctionne pas

Flux affecté :

Adresse IP source : 192.168.0.100

Adresse IP de destination : 10.10.1.100

Protocole : TCP 80

Analyse de capture

Activez les captures sur le moteur FTD LINA.

<#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



Captures - Scénario non fonctionnel :

Voici à quoi ressemblent les captures à partir de l'interface de ligne de commande du périphérique :

```
<#root>
firepower#
show capture
capture CAPI type raw-data trace interface INSIDE [Capturing -
834 bytes
٦
 match ip host 192.168.0.100 host 10.10.1.100
capture CAPO type raw-data interface OUTSIDE [Capturing -
878 bytes
٦
 match ip host 192.168.0.100 host 10.10.1.100
Contenu CAPI :
<#root>
firepower#
show capture CAPI
   1: 05:20:36.654217 192.168.0.100.22195 > 10.10.1.100.80:
s
1397289928:1397289928(0) win 8192 <mss 1460,nop,wscale 2,nop,nop,sackOK>
                      192.168.0.100.22196 > 10.10.1.100.80:
   2: 05:20:36.904311
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>
                      10.10.1.100.80 > 192.168.0.100.22196:
   5: 05:20:37.414803
R
31997177:31997177(0) ack 2171673259 win 0
   6: 05:20:37.914183 192.168.0.100.22196 > 10.10.1.100.80:
s
```

Contenu CAPO :

<#root>

firepower#

show capture CAPO

1: 05:20:36.654507 802.1Q vlan#202 P0 192.168.0.100.22195 > 10.10.1.100.80: s 2866789268:2866789268(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>

```
2: 05:20:36.904478 802.1Q vlan#202 P0 192.168.0.100.22196 > 10.10.1.100.80:

s
4785344:4785344(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>

3: 05:20:36.904997 802.1Q vlan#202 P0 10.10.1.100.80 > 192.168.0.100.22196:

R
0:0(0) ack 4785345 win 0

4: 05:20:37.414269 802.1Q vlan#202 P0 192.168.0.100.22196 > 10.10.1.100.80:

s
4235354730:4235354730(0) win 8192 <mss 1380,nop,wscale 2,nop,nop,sackOK>

5: 05:20:37.414758 802.1Q vlan#202 P0 10.10.1.100.80 > 192.168.0.100.22196:

R
```

```
0:0(0) ack 4235354731 win 0
6: 05:20:37.914305 802.1Q vlan#202 P0 192.168.0.100.22196 > 10.10.1.100.80:
s
4118617832:4118617832(0) win 8192 <mss 1380,nop,nop,sackOK>
```

Cette image montre la capture de CAPI dans Wireshark.

_								
No.	Time	Source	Destination	Protocol Length	Info			
	10.000000	192.168.0.100	10.10.1.100	TCP 66	22195 → 80 [SYN]=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1			
	2 0.250094	192.168.0.100	10.10.1.100	TCP 66	22196 → 80 [SYN] = 4=0 Win=8192 Len=0 MSS=1460 HS=4 SACK_PERM=1			
	3 0.000732	10.10.1.100	192.168.0.100	TCP 54	80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 📿			
	4 0.509089	192.168.0.100	10.10.1.100	тср 3	[TCP Retransmission] 22196 \rightarrow 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1			
	5 0.000671	10.10.1.100	192.168.0.100	тср 54	80 → 22196 [RST, ACK] Seq=2476911971 Ack=1 Win=0 Len=0			
	6 0.499380	192.168.0.100	10.10.1.100	TCP 62	[TCP Retransmission] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1			
	7 0.000625	10.10.1.100	192.168.0.100	TCP 54	80 → 22196 [RST, ACK] Seq=2853655305 Ack=1 Win=0 Len=0			
	8 1.739729	192.168.0.100	10.10.1.100	TCP 66	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1			
	9 0.000611	10.10.1.100	192.168.0.100	TCP 54	80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0			
	10 0.499385	192.168.0.100	10.10.1.100	TCP 62	[TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1			
L	11 0.000671	10.10.1.100	192.168.0.100	TCP 54	80 → 22195 [RST, ACK] Seq=151733665 Ack=1 Win=0 Len=0			
-								
>	> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)							
>	Ethernet II,	<pre>Src: Cisco_fc:f</pre>	c:d8 (4c:4e:35:f	fc:fc:d8), Dst:	Cisco_f6:1d:ae (00:be:75:f6:1d:ae 4			
>	Internet Prot	cocol Version 4,	Src: 192.168.0.	100, Dst: 10.1	0.1.100			
>	Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Seg: 0, Len: 0							

Principaux points :

- 1. La source envoie un paquet TCP SYN.
- 2. Un RST TCP est envoyé vers la source.
- 3. La source retransmet les paquets TCP SYN.
- 4. Les adresses MAC sont correctes (sur les paquets entrants, l'adresse MAC source appartient au routeur en aval, l'adresse MAC de destination appartient à l'interface INSIDE du pare-feu).

Cette image montre la capture de CAPO dans 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] 100 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1					
	2 2019-10-11 07:20:36.904478 192.168.0.	100 10.10.1.100	TCP	70 22196 \rightarrow 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1					
	3 2019-10-11 07:20:36.904997 10.10.1.10	0 192.168.0.100	TCP	58 80 → 22196 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 2					
	4 2019-10-11 07:20:37.414269 192.168.0.	100 10.10.1.100	тср	70 [TCP Port numbers reused] 22196 → 80 [SYN] Seq=741 C Port numbers reused] 22196 → 80 [SYN] Seq=741 C Port numbers reused]					
	5 2019-10-11 07:20:37.414758 10.10.1.10	0 192.168.0.100	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	тср	66 [TCP Port numbers reused] 22196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 SACK_PERM=1					
	7 2019-10-11 07:20:37.914762 10.10.1.10	0 192.168.0.100	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	тср	70 [TCP Retransmission] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1					
	9 2019-10-11 07:20:39.655102 10.10.1.10	0 192.168.0.100	TCP	58 80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0					
L	10 2019-10-11 07:20:40.154700 192.168.0.	100 10.10.1.100	тср	66 [TCP Port numbers reused] 22195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 SACK_PERM=1					
	11 2019-10-11 07:20:40.155173 10.10.1.10	0 192.168.0.100	TCP	58 80 → 22195 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0					
<									
>	> Frame 1: 70 bytes on wire (560 bits), 70 bytes cap 4 (560 bits)								
>	Ethernet II, Src: Cisco_f6:1d:8e (00:be:	75:f6:1d:8e), Dst: Cisco	_fc:fc:d8	8 (4c:4e:35:fc:fc:d8)					
>	802.10 Virtual LAN, PRI: 0, DEI: 0, ID: 3	202							
>	Internet Protocol Version 4, Src: 192.16	8.0.100, Dst: 10.10.1.10	0						
>	> Transmission Control Protocol, Src Port: 22195, Dst Port: 80, Sea: 0, Len: 0								
· · ·									

Principaux points :

- 1. La source envoie un paquet TCP SYN.
- 2. Un RST TCP arrive sur l'interface OUTSIDE.
- 3. La source retransmet les paquets TCP SYN.
- 4. Les adresses MAC sont correctes (sur les paquets de sortie, le pare-feu OUTSIDE est l'adresse MAC source, le routeur en amont est l'adresse MAC de destination).

Sur la base des deux captures, on peut conclure que :

- La connexion TCP en trois étapes entre le client et le serveur n'est pas terminée
- Il y a un TCP RST qui arrive sur l'interface de sortie du pare-feu
- Le pare-feu « communique » avec les périphériques en amont et en aval appropriés (en fonction des adresses MAC)

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Vérifiez l'adresse MAC source qui envoie la RST TCP.

Vérifiez que l'adresse MAC de destination vue dans le paquet TCP SYN est identique à l'adresse MAC source vue dans le paquet TCP RST.

CAPO_RST_SERVER.pcap									
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics Telephon <u>y W</u> ireless <u>T</u> ools <u>H</u> elp									
🚄 🔳 🖉 🔍 📕 🛅 🕱 🚱 🍳 🖛 🌧 🖀 7 👲 🔂 🔲 Q. Q. Q. 11									
Apply a display filter <ctrl-></ctrl->									
No. Time Source Destination Protocol Length Info	io								
1 2019-10-11 07:20:36.654507 192.168.0.100 10.10.1.100 TCP 70 22	195 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1								
□ 2 2019-10-11 07:20:36.904478 192.168.0.100 10.10.1.100 TCP 70 22	196 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1								
> Frame 2: 70 bytes on wire (560 bits) 70 bytes cantured (560 bits)									
<pre>> Ethernet II, Src: Cisco_f6:1d:8e (00:be:75:f6:1d:8e) Dst: Cisco_fc:fc:d8 (4c:4e:35: > 002 40 Vistual 4</pre>	:fc:fc:d8)								
> 802.10 Virtual LAW, PKI: 0, DEI: 0, 1D: 202									
> Transmission Control Protocol, Src Port: 22196, DS, Port: 80, Sec. 0, Len: 0									
CAPO_RST_SERVER.pcap									
File Edit View, Co. Conture Analyza Statistica Telenhany, Wizeley Tean, Hele									
The Fort Alexandre Busines Freedom Analyze Statistics leichbourd Miller Tool Helb									
Apply a display hiter < Ctrl-/>									
No. Time Source Destination Protocol Length Info									
1 2019-10-11 07:20:36.654507 192.168.0.100 10.10.1.100 P 70 22	$2195 \rightarrow 80$ [SYN] Seq=0 Win=8192 Len=0 MSS=1380 WS=4 SACK_PERM=1								
3 2019-10-11 07:20:36 904997 10 10 1 100 192 168 0 100 TCP 58 80	$190 \rightarrow 30$ [STN] SEQ=0 WIN=8192 LEN=0 MSS=1380 WS=4 SACK_PERM=1								
<									
> Frame 3: 58 bytes on wire (464 bits) 58 bytes cantured (464 bits)									
> Frame 3: 58 bytes on wire (464 hits) 58 bytes cantuced (464 bits) > Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8) Dst: Cisco f6:1d:8e (00:be:75:	:f6:1d:8e)								
 > Frame 3: 58 bytes on wire (464 hits) 58 bytes cantuced (464 bits) > Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8) Dst: Cisco_f6:1d:8e (00:be:75: > 802.1Q Virtual LAN, РКІ: 0, DEI: 0, 1D: 202 	:f6:1d:8e)								
 > Frame 3: 58 bytes on wire (464 hits) 58 bytes cantured (464 bits) > Ethernet II, Src: Cisco_fc:fc:d8 (4c:4e:35:fc:fc:d8) Dst: Cisco_f6:1d:8e (00:be:75: > 802.1Q Virtual LAN, PRI: 0, DEI: 0, 1D: 202 > Internet Protocol Version 4, Src: 10.10.1.100, Dst: 192.168.0.100 	:f6:1d:8e)								

Cette vérification a pour but de confirmer 2 choses :

- Vérifiez qu'il n'y a pas de flux asymétrique.
- Vérifiez que l'adresse MAC appartient au périphérique en amont attendu.

Action 2. Comparer les paquets entrants et sortants.

Comparez visuellement les 2 paquets sur Wireshark pour vérifier que le pare-feu ne modifie pas/ne corrompt pas les paquets. Certaines différences attendues sont mises en évidence.



Principaux points :

- 1. Les horodatages sont différents. D'un autre côté, la différence doit être faible et raisonnable. Cela dépend des fonctions et des contrôles de stratégie appliqués au paquet ainsi que de la charge sur le périphérique.
- 2. La longueur des paquets diffère en particulier si un en-tête dot1Q est ajouté/supprimé par le pare-feu sur un seul côté.
- 3. Les adresses MAC sont différentes.
- 4. Un en-tête dot1Q peut être en place si la capture a été effectuée sur une sous-interface.
- 5. Les adresses IP sont différentes si la traduction d'adresses de port (PAT) ou la traduction d'adresses de port (NAT) est appliquée au paquet.
- 6. Les ports source ou de destination sont différents si la fonction NAT ou PAT est appliquée au paquet.
- 7. Si vous désactivez l'option Wireshark Relative Sequence Number, vous voyez que les numéros de séquence TCP/les numéros d'accusé de réception sont modifiés par le pare-feu en raison de la randomisation ISN (Initial Sequence Number).
- 8. Certaines options TCP peuvent être remplacées. Par exemple, le pare-feu modifie par défaut la taille maximale de segment (MSS) TCP sur 1380 afin d'éviter la fragmentation des paquets dans le chemin de transit.

Action 3. Effectuez une capture à destination.

Si possible, effectuez une capture à la destination elle-même. Si ce n'est pas possible, effectuez une capture aussi près que possible de la destination. L'objectif ici est de vérifier qui envoie la RST TCP (le serveur de destination ou un autre périphérique se trouve-t-il sur le chemin ?).

Cas 3 . Connexion TCP en trois étapes + RST à partir d'un terminal



Cette image présente la topologie :

Description du problème : HTTP ne fonctionne pas

Flux affecté :

Adresse IP source : 192.168.0.100

Adresse IP de destination : 10.10.1.100

Protocole : TCP 80

Analyse de capture

Activez les captures sur le moteur FTD LINA.

<#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



Captures - Scénario non fonctionnel :

Ce problème peut se manifester de deux façons différentes dans les captures.

3.1 - Connexion TCP en trois étapes + RST différé du client

Les captures CAPI et CAPO du pare-feu contiennent les mêmes paquets, comme illustré dans l'image.

No.	Time So	iource	Destination	Protocol I	ength Info
ŕ	2 2019-10-13 17:06:27.874085 19	92.168.0.100	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	0.10.1.100	192.168.0.100	TCP	66 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
	4 2019-10-13 17:06:27.875183 19	92.168.0.100	10.10.1.100	ТСР	54 48295 → 80 [ACK] Seq=179631562 Ack=3838911938 Win=66240 Len=0
Τ	8 2019-10-13 17:06:30.882537 10		192.168.0.100	ТСР 🧲	66 [TCP Retransmission] 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=8192 Len=0 MSS=1380 WS=256 SACK_PERM=1
	9 2019-10-13 17:06:30.883056 19	92.168.0.100		тср 🚬	🚩 66 [TCP Previous segment not captured] 48295 → 80 [ACK] Seq=179631962 Ack=3838911938 Win=66240 Len=0 SLE=3838911937 SRE=3838911938
	13 2019-10-13 17:06:36.889022 10		192.168.0.100	ТСР	62 [TCP Retransmission] 80 → 48295 [SYN, ACK] Seq=3838911937 Ack=179631562 Win=65535 Len=0 MSS=1380 SACK_PERM=1
		92.168.0.100		ТСР 🧲	66 [TCP Dup ACK 4#1] 48295 → 80 [ACK] Seq=179631962 Ack=3838911938 Win=66240 Len=0 SLE=3838911937 SRE=3838911938
L	17 2019-10-13 17:06:47.943631 19	92.168.0.100	10.10.1.100	TCP	54 48295 → 80 [RST, ACK] Seq=179631962 Ack=3838911938 Win=0 Len=0

Principaux points :

- 1. La connexion TCP en trois étapes passe par le pare-feu.
- 2. Le serveur retransmet le message SYN/ACK.
- 3. Le client retransmet l'accusé de réception.
- 4. Après environ 20 secondes, le client abandonne et envoie un RST TCP.

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Effectuez des captures aussi près que possible des deux terminaux.

Les captures du pare-feu indiquent que le serveur n'a pas traité l'ACK du client. Ceci est basé sur les faits suivants :

- Le serveur retransmet le message SYN/ACK.
- Le client retransmet l'accusé de réception.
- Le client envoie un RST TCP ou un FIN/ACK avant toute donnée.

La capture sur le serveur montre le problème. Le client ACK de la connexion TCP en trois étapes n'est jamais arrivé :

- 26 7.636612 192.168.0.100 10.10.1.100 TCP 66 55324+80 [SYN] Seq=433201323 Win=8192 Len=0 MSS=1380 WS=4 SAC 29 7.637571 10.10.1.100 192.168.0.100 TCP 66 80+55324 [SYN] Seq=4063222169 Ack=433201324 Win=8192 Len 30 7.930152 192.168.0.100 10.10.1.100 TCP 66 55325+80 [SYN] Seq=366197499 Win=8192 Len=0 MSS=1380 WS=4 SAC 31 7.930221 10.10.1.100 192.168.0.100 TCP 66 80+55325 [SYN, ACK] Seq=2154790336 Ack=366197500 Win=8192 Len 41 10.629868 192.168.0.100 10.10.1.100 TCP 66 [CCP Spurious Retransmission] 55324+80 [SYN] Seq=43201323 Win 42 10.633208 10.10.1.100 TCP 66 [CCP Retransmission] 80+55324 [SYN, ACK] Seq=4063222169 Ack=432013232						
29 7.637571 10.10.1.100 192.168.0.100 TCP 66 80+55324 [SYN, ACK] Seq=4063222169 Ack=433201324 Win=8192 Len 30 7.930152 192.168.0.100 10.10.1.100 TCP 66 55325+80 [SYN] Seq=366197499 Win=8192 Len=0 MSS=1380 WS=4 SAC 31 7.930221 10.10.1.100 192.168.0.100 TCP 66 80+55325 [SYN, ACK] Seq=2154790336 Ack=366197500 Win=8192 Len 41 10.629868 192.168.0.100 10.10.1.100 TCP 66 [CTCP Spurious Retransmission] 55324+80 [SYN] Seq=4063222169 Ack=34063222169 Ack=43201323 Win=8192 Len 42 10.633208 10.10.1.100 192.168.0.100 TCP 66 [TCP Retransmission] 80+55324 [SYN, ACK] Seq=4063222169 Ack=43201323 Win=8192 Len	-3	26 7.636612	192.168.0.100	10.10.1.100	TCP	66 55324→80 [SYN] Seq=433201323 Win=8192 Len=0 MSS=1380 WS=4 SAC…
30 7.930152 192.168.0.100 10.10.1.100 TCP 66 55325+80 [SYN] Seq=366197499 Win=8192 Len=0 MSS=1380 WS=4 SAC 31 7.930221 10.10.1.100 192.168.0.100 TCP 66 80+55325 [SYN, ACK] Seq=2154790336 Ack=366197500 Win=8192 Len 41 10.629868 192.168.0.100 10.10.1.100 TCP 66 [CCP Spurious Retransmission] 55324+80 [SYN] Seq=433201323 Win=4063222169 Ack=4663222169 Ack=4663222169		29 7.637571	10.10.1.100	192.168.0.100	TCP	66 80→55324 [SYN, ACK] Seq=4063222169 Ack=433201324 Win=8192 Len…
31 7.930221 10.10.1.100 192.168.0.100 TCP 66 80+55325 [SYN, ACK] Seq=2154790336 Ack=366197500 Win=8192 Ler 41 10.629868 192.168.0.100 10.10.1.100 TCP 66 [CCP Spurious Retransmission] 55324+80 [SYN] Seq=433201323 Win 42 10.633208 10.10.1.100 TCP 66 [TCP Retransmission] 80+55324 [SYN, ACK] Seq=4063222169 Ack=4		30 7.930152	192.168.0.100	10.10.1.100	TCP	66 55325→80 [SYN] Seq=366197499 Win=8192 Len=0 MSS=1380 WS=4 SAC…
41 10.629868 192.168.0.100 10.10.1.100 TCP 66 [TCP Spurious Retransmission] 55324+80 [SYN] Seq=433201323 Wi 42 10.633208 10.10.1.100 192.168.0.100 TCP 66 [TCP Retransmission] 80+55324 [SYN, ACK] Seq=4063222169 Ack=4		31 7.930221	10.10.1.100	192.168.0.100	ТСР	66 80→55325 [SYN, ACK] Seq=2154790336 Ack=366197500 Win=8192 Len…
42 10.633208 10.10.1.100 192.168.0.100 TCP 66 [TCP Retransmission] 80→55324 [SYN, ACK] Seq=4063222169 Ack=4		41 10.629868	192.168.0.100	10.10.1.100	тср	66 [TCP Spurious Retransmission] 55324→80 [SYN] Seq=433201323 Wi…
		42 10.633208	10.10.1.100	192.168.0.100	тср	66 [TCP Retransmission] 80→55324 [SYN, ACK] Seq=4063222169 Ack=4…
44 10.945178 10.10.1.100 192.168.0.100 TCP 66 [TCP Retransmission] 80→55325 [SYN, ACK] Seq=2154790336 Ack=3		44 10.945178	10.10.1.100	192.168.0.100	ТСР	66 [TCP Retransmission] 80→55325 [SYN, ACK] Seq=2154790336 Ack=3…
60 16.636255 192.168.0.100 10.10.1.100 TCP 62 [TCP Spurious Retransmission] 55324+80 [SYN] Seq=433201323 Wi	ľ	60 16.636255	192.168.0.100	10.10.1.100	тср	62 [TCP Spurious Retransmission] 55324→80 [SYN] Seq=433201323 Wi…
61 16.639145 10.10.1.100 192.168.0.100 TCP 62 [TCP Retransmission] 80+55324 [SYN, ACK] Seq=4063222169 Ack=4		61 16.639145	10.10.1.100	192.168.0.100	тср	62 [TCP Retransmission] 80→55324 [SYN, ACK] Seq=4063222169 Ack=4
62 16.951195 10.10.1.100 192.168.0.100 TCP 62 [TCP Retransmission] 80+55325 [SYN, ACK] Seq=2154790336 Ack=3		62 16.951195	10.10.1.100	192.168.0.100	ТСР	62 [TCP Retransmission] 80+55325 [SYN, ACK] Seq=2154790336 Ack=3

3.2 - Connexion TCP en trois étapes + FIN/ACK retardé du client + RST retardé du serveur

Les captures CAPI et CAPO du pare-feu contiennent les mêmes paquets, comme illustré dans l'image.


Principaux points :

- 1. La connexion TCP en trois étapes passe par le pare-feu.
- 2. Après ~5 secondes, le client envoie un FIN/ACK.
- 3. Après ~20 secondes, le serveur abandonne et envoie un RST TCP.

Sur la base de cette capture, on peut conclure que bien qu'il y ait une connexion TCP en trois étapes à travers le pare-feu, il semble qu'elle ne soit jamais réellement terminée sur un point d'extrémité (les retransmissions l'indiquent).

Actions recommandées

Identique au cas 3.1

3.3 - Connexion TCP en trois étapes + RST différé du client

Les captures CAPI et CAPO du pare-feu contiennent les mêmes paquets, comme illustré dans l'image.

No.		Time	Source	Destination	Protocol	Length	Info
Г	129	2019-10-13 17:09:20.513355	192.168.0.100	10.10.1.100	ТСР	6	6 48355 → 80 [SYN] Seq=2581697538 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
	130	2019-10-13 17:09:20.514011	10.10.1.100	192.168.0.100	ТСР	6	6 80 → 48355 [SYN, ACK] Seq=1633018698 Ack=2581697539 Win=8192 Len=0 MSS=1
	131	2019-10-13 17:09:20.514438	192.168.0.100	10.10.1.100	ТСР	54	4 48355 → 80 [ACK] Seq=2581697539 Ack=1633018699 Win=66240 Len=0
L	132	2019-10-13 17:09:39.473089	192.168.0.100	10.10.1.100	TCP	2 5	4 48355 → 80 [RST, ACK] Seq=2581697939 Ack=1633018699 Win=0 Len=0

Principaux points :

- 1. La connexion TCP en trois étapes passe par le pare-feu.
- 2. Après environ 20 secondes, le client abandonne et envoie un RST TCP.

Sur la base de ces captures, on peut conclure que :

• Après 5 à 20 secondes, un terminal abandonne et décide de mettre fin à la connexion.

Actions recommandées

Identique au cas 3.1

3.4 - Connexion TCP en trois étapes + RST immédiat à partir du serveur

Les captures CAPI et CAPO du pare-feu contiennent ces paquets, comme illustré dans l'image.

No.		Time	Source	Destination	Protocol	Length	Info					
Г	26	2019-10-13 17:07:07.104410	192.168.0.100	10.10.1.100	TCP	66	48300 → 80	[SYN]	Seq=2563435279	Win=8192 Len=0	MSS=1460 WS=4	SACK_PERM=1
	27	2019-10-13 17:07:07.105112	10.10.1.100	192.168.0.100	TCP	66	80 → 48300	[SYN,	ACK] Seq=37571	37497 Ack=25634	35280 Win=8192	Len=0 MSS=1380
	28	2019-10-13 17:07:07.105554	192.168.0.100	10.10.1.100	TCP	54	48300 → 80	[ACK]	Seq=2563435286	Ack=3757137498	Win=66240 Len=	=0
L	41	2019-10-13 17:07:07.106325	10.10.1.100	192.168.0.100	ТСР	54	80 → 48300	[RST]	Seq=2563435286	0 Win=0 Len=0		

Principaux points :

- 1. La connexion TCP en trois étapes passe par le pare-feu.
- 2. Il y a un RST TCP du serveur quelques millisecondes après le paquet ACK.

Actions recommandées

Action : effectuez des captures aussi près que possible du serveur.

Un RST TCP immédiat provenant du serveur peut indiquer un serveur défaillant ou un périphérique sur le chemin qui envoie le RST TCP. Effectuez une capture sur le serveur lui-même et déterminez la source du RST TCP.

Cas 4 . TCP RST à partir du client

Cette image présente la topologie :



Description du problème : HTTP ne fonctionne pas.

Flux affecté :

Adresse IP source : 192.168.0.100

Adresse IP de destination : 10.10.1.100

Protocole : TCP 80

Analyse de capture

Activer les captures sur le moteur FTD LINA.

<#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



Captures - Scénario non fonctionnel :

Il s'agit du contenu CAPI.

<#root>

firepower#

show capture CAPI

14 packets captured

1:	12:32:22.860627	192.168.0.100.47078 >	10.10.1.100.80:	S	4098574664:4098574664(0)	win	8192	<mss< th=""></mss<>
2:	12:32:23.111307	192.168.0.100.47079 >	10.10.1.100.80:	S	2486945841:2486945841(0)	win	8192	<mss< td=""></mss<>
3:	12:32:23.112390	192.168.0.100.47079 >	10.10.1.100.80:	R	3000518858:3000518858(0)	win	0	
4:	12:32:25.858109	192.168.0.100.47078 >	10.10.1.100.80:	S	4098574664:4098574664(0)	win	8192	<mss< td=""></mss<>
5:	12:32:25.868698	192.168.0.100.47078 >	10.10.1.100.80:	R	1386249853:1386249853(0)	win	0	
6:	12:32:26.108118	192.168.0.100.47079 >	10.10.1.100.80:	S	2486945841:2486945841(0)	win	8192	<mss< td=""></mss<>
7:	12:32:26.109079	192.168.0.100.47079 >	10.10.1.100.80:	R	3000518858:3000518858(0)	win	0	
8:	12:32:26.118295	192.168.0.100.47079 >	10.10.1.100.80:	R	3000518858:3000518858(0)	win	0	
9:	12:32:31.859925	192.168.0.100.47078 >	10.10.1.100.80:	S	4098574664:4098574664(0)	win	8192	<mss< td=""></mss<>
10:	12:32:31.860902	192.168.0.100.47078 >	10.10.1.100.80:	R	1386249853:1386249853(0)	win	0	
11:	12:32:31.875229	192.168.0.100.47078 >	10.10.1.100.80:	R	1386249853:1386249853(0)	win	0	
12:	12:32:32.140632	192.168.0.100.47079 >	10.10.1.100.80:	R	3000518858:3000518858(0)	win	0	
13:	12:32:32.159995	192.168.0.100.47079 >	10.10.1.100.80:	S	2486945841:2486945841(0)	win	8192	<mss< td=""></mss<>
14:	12:32:32.160956	192.168.0.100.47079 >	10.10.1.100.80:	R	3000518858:3000518858(0)	win	0	
14 pa	ckets shown							

Voici le contenu du CAPO :

<#root>

firepower#

show capture CAPO

11 packets captured

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

Les journaux du pare-feu affichent :

<#root>

firepower#

show log | i 47741

Oct 13 2019 13:57:36: %FTD-6-302013: Built inbound TCP connection 4869 for INSIDE:192.168.0.100/47741 (Oct 13 2019 13:57:36: %FTD-6-302014: Teardown TCP connection 4869 for INSIDE:192.168.0.100/47741 to OUT

TCP Reset-O from INSIDE

Oct 13 2019 13:57:39: %FTD-6-302013: Built inbound TCP connection 4870 for INSIDE:192.168.0.100/47741 (Oct 13 2019 13:57:39: %FTD-6-302014: Teardown TCP connection 4870 for INSIDE:192.168.0.100/47741 to OUT

TCP Reset-O from INSIDE

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

Ces journaux indiquent qu'un TCP RST arrive sur l'interface INSIDE du pare-feu

Capture CAPI dans Wireshark :

Suivez le premier flux TCP, comme illustré dans l'image.

No.	Time Source	Destination	Protocol Length	h Info		
Г	1 2019-10-13 14:32:22.860627 192.168.0.100 2 2019-10-13 14:32:23.111307 192.168.0.100	10.10.1.100 10.10.1.100	ТСР ТСР	66 47078 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PE 66 47079 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PE	Mark/Unmark Packet	
	3 2019-10-13 14:32:23.112390 192.168.0.100 4 2019-10-13 14:32:25.858109 192.168.0.100 5 2019-10-13 14:32:25.868698 192.168.0.100	10.10.1.100 10.10.1.100 10.10.1.100	TCP TCP TCP	54 47079 + 80 [RST] Seq=513573017 Win=0 Len=0 66 [TCP Retransmission] 47078 + 80 [SYN] Seq=0 Win=8192 Len=0 54 47078 + 80 [RST] Seq=1582642485 Win=0 Len=0	Ignore/Unignore Packet Set/Unset Time Reference Time Shift	
	6 2019-10-13 14:32:26.108118 192.168.0.100 7 2019-10-13 14:32:26.109079 192.168.0.100	10.10.1.100	ТСР	66 [TCP Retransmission] 47079 → 80 [SYN] Seq=0 Win=8192 Len=0 54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0	Packet Comment	
	8 2019-10-13 14:32:26.118295 192.168.0.100 9 2019-10-13 14:32:31.859925 192.168.0.100	10.10.1.100	ТСР	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0 62 [TCP Retransmission] 47078 → 80 [SYN] Seq=0 Win=8192 Len=0	Apply as Filter	
L	10 2019-10-13 14:32:31.860902 192.168.0.100 11 2019-10-13 14:32:31.875229 192.168.0.100	10.10.1.100	ТСР ТСР	54 47078 → 80 [RST] Seq=1582642485 Win=0 Len=0 54 47078 → 80 [RST] Seq=1582642485 Win=0 Len=0	Prepare a Filter	
	12 2019-10-13 14:32:32.140632 192.168.0.100 13 2019-10-13 14:32:32.159995 192.168.0.100	10.10.1.100 10.10.1.100	ТСР ТСР	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0 62 [TCP Retransmission] 47079 → 80 [SYN] Seq=0 Win=8192 Len=0	Colorize Conversation	
	14 2019-10-13 14:32:32.160956 192.168.0.100	10.10.1.100	тср	54 47079 → 80 [RST] Seq=513573017 Win=0 Len=0	Follow	TCP Stream
					Сору	UDP Stream SSL Stream
					Protocol Preferences Decode As	HTTP Stream
					Show Packet in New Window	

Sous Wireshark, accédez à Edit > Preferences > Protocols > TCP et désélectionnez l'option Relative sequence numbers comme indiqué dans l'image.

Wireshark - Preferences	?	×
Steam IHS D Transmission Control Protocol STP Show TCP summary in protocol tree STUN Validate the TCP checksum if possible SUA Allow subdissector to reassemble TCP streams SV Analyze TCP sequence numbers SYNC Relative sequence numbers Synergy Syslog T.38 Calculate conversation timestamps TACACS Try heuristic sub-dissectors first TACACS+ Try heuristic sub-dissectors for error packets TCP Do not call subdissectors for error packets TCP Display process information via IPFEX TCP UDP port 0		~
OK Cancel	Help	

Cette image montre le contenu du premier flux dans la capture CAPI :

	xp.stream eq 0						_			
No.	Time	Source	Destination	Protocol Length	Info					
E.	1 2019-10-13 14:32:22.860627	192.168.0.100	10.10.1.100	TCP	66 47078 ÷	80 [SYN] Se	eq=4098574664	Win=8192 Len=	0 MSS=146	0 WS=4 SACK_PERM=1
	4 2019-10-13 14:32:25.858109	192.168.0.100	10.10.1.100	TCP	66 [TCP Ret	ransmission	n] 47078 → 80	[SYN] Seq=409	8574664 W	lin=8192 Len=0 MSS=1
	5 2019-10-13 14:32:25.868698	\$ 192.168.0.100	10.10.1.100	TCP	54 47078 + 1	80 [RST] Se	eq=1386249853	Win=0 Len=0	2)	
	9 2019-10-13 14:32:31.859925	5 192.168.0.100	10.10.1.100	TCP (62 [TCP Ret	ransmission	n] 47078 → 80	[SYN] Seq=409	8574664 W	lin=8192 Len=0 MSS=1
	10 2019-10-13 14:32:31.860902	192.168.0.100	10.10.1.100	TCP	54 47078 🔸	80 [RST] So	eq=1386249853	Win=0 Len=0		
L	11 2019-10-13 14:32:31.875229	192.168.0.100	10.10.1.100	TCP	54 47078 +	80 [RST] Se	eq=1386249853	Win=0 Len=0		
1										
<u> </u>										
>	Frame 1: 66 bytes on wire (528	3 bits), 66 byte	s captured (528	bits)						
2	Ethernet II, Src: Cisco_fc:fc:	:d8 (4c:4e:35:fc	::fc:d8), Dst: Ci	sco_f6:1d:ae	(00:be:75:	f6:1d:ae)				
>	Internet Protocol Version 4, 5	Src: 192.168.0.1	00, Dst: 10.10.1	.100		-				
~	Transmission Control Protocol,	, Src Port: 4707	78, Dst Port: 80,	Seq: 409857	4664, Len:	0				
	Source Port: 47078									
	Destination Port: 80									
	[Stream index: 0]	•								
	[ICP Segment Len: 0]	(3)								
	Sequence number: 4098574664									
	[Next sequence number: 4098	574004 J								
	Acknowledgment number: 0	22 huter (8)								
	Elage: 0x002 (SVN)	32 Dytes (8)								
	Vinder size value: 2102									
	[(a)culated window size: 01	021								
	Chacksum: 0x8cd1 [unuarifia	.92 J								
	[Chacksum Status: Invarifia	41								
	Urgent pointer: 0									
	> Ontions: (12 hytes) Mavimu	m segment size	No-Operation (N	(P) Window	scale No.0	neration (NOP) No-Oner-	ation (NOP). S	ACK nermi	itted
) [Timestamos]	m selfmente streft	the operation (in	, , , , , , , , , , , , , , , , , , ,	, 10 ⁻⁰	ber across (t	iter /) no open	(inter); 3	new permi	
L	- [.mestades]									

Principaux points :

- 1. Le client envoie un paquet TCP SYN.
- 2. Le client envoie un paquet TCP RST.
- 3. Le paquet TCP SYN a une valeur de numéro d'ordre égale à 4098574664.

Le même flux de capture CAPO contient :

No.		Time	Source	Destination	Protocol Length	Info							
Г	1	2019-10-13 14:32:22.860780	192.168.0.100	10.10.1.100	ТСР	70 47078 → 80 [SYN] Seq=1386249852 N8192 Len=0 MSS=1380 WS=4 SACK_PERM=1							
	4	2019-10-13 14:32:25.858125	192.168.0.100	10.10.1.100	тср	70 [TCP Retransmission] 47078 → 80 [SYN] Seq=1386249852 Win=8192 Len=0 MSS=1380							
	5 2019-10-13 14:32:25.868729 192.168.0.100 10.10.1.100 TCP 58 47078 → 80 [RST] Seq=2968892337 Win=0 Len=0												
	2												
<													
>	> 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												

Principaux points :

- 1. Le client envoie un paquet TCP SYN. Le pare-feu randomise l'ISN.
- 2. Le client envoie un paquet TCP RST.

Sur la base des deux captures, on peut conclure que :

- Il n'y a pas de connexion TCP en trois étapes entre le client et le serveur.
- Il y a un TCP RST qui vient du client. La valeur du numéro de séquence TCP RST dans la capture CAPI est 1386249853.

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Effectuez une capture du client.

Sur la base des captures collectées sur le pare-feu, il existe une forte indication d'un flux asymétrique. Ceci est basé sur le fait que le client envoie un TCP RST avec une valeur de 1386249853 (le RNIS aléatoire) :

Image: Provide state st	No.	Time	Source	Destination	Protocol	Length Info
29 9.037499 192.168.0.100 10.10.1.100 TCP 2 66 [TCP Retransmission] 47078+80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1 30 9.048155 10.10.1.100 192.168.0.100 TCP 2 66 [TCP ACKed unseen segment] 80+47078 [SYN, ACK] Seq=1924342422 Ack=13862	Г	19 6.040337	192.168.0.100	10.10.1.100	TCP	66 47078→80 [SYN] Seq=4098574664 92 Len=0 MSS=1460 WS=4 SACK_PERM=1
30 9.048155 10.10.1.100 192.168.0.100 TCP 66 [TCP ACKed unseen segment] 80+47078 [SYN, ACK] Seq=1924342422 Ack=13862		29 9.037499	192.168.0.100	10.10.1.100	ТСР	66 [TCP Retransmission] 47078→80 [SYN] Seq=4098574664 Win=8192 Len=0 MSS=1460 WS
21 0 049194 102 169 0 100 10 10 1 100 TCD E4 47079 90 [DCT] Con-1296240952 Uin-0 Lon-0		30 9.048155	10.10.1.100	192.168.0.100	ТСР	66 [TCP ACKed unseen segment] 80→47078 [SYN, ACK] Seq=1924342422 Ack=1386249853 N
- 51 9.040104 192.100.0.100 10.10.1.100 ICP 54 4/0/0400 [K51] Seq=1500249055 Win=0 Len=0	L.	31 9.048184	192.168.0.100	10.10.1.100	TCP	54 47078→80 [RST] Seq=1386249853 Win=0 Len=0 🤦

Principaux points :

- 1. Le client envoie un paquet TCP SYN. Le numéro d'ordre est 4098574664 et est le même que celui vu sur l'interface INSIDE du pare-feu (CAPI)
- 2. Il existe un TCP SYN/ACK avec ACK numéro 1386249853 (ce qui est attendu en raison de la randomisation ISN). Ce paquet n'a pas été vu dans les captures du pare-feu
- 3. Le client envoie un RST TCP car il attendait un SYN/ACK avec une valeur de numéro ACK de 4098574665, mais il a reçu une valeur de 1386249853

Cela peut être visualisé comme suit :



Action 2. Vérifiez le routage entre le client et le pare-feu.

Confirmez que :

- Les adresses MAC visibles dans les captures sont les adresses attendues.
- Vérifiez que le routage entre le pare-feu et le client est symétrique.

Il existe des scénarios où la TVD provient d'un périphérique situé entre le pare-feu et le client alors qu'il existe un routage asymétrique dans le réseau interne. Un cas typique est montré dans l'image :



Dans ce cas, la capture a ce contenu. Notez la différence entre l'adresse MAC source du paquet TCP SYN et l'adresse MAC source du RST TCP et l'adresse MAC de destination du paquet TCP SYN/ACK :

<#root>

firepower#

show capture CAPI detail

1: 13:57:36.730217

4c4e.35fc.fcd8

```
00be.75f6.1dae 0x0800 Length: 66
    192.168.0.100.47740 > 10.10.1.100.80: S [tcp sum ok] 3045001876:3045001876(0) win 8192 <mss 1460,
2: 13:57:36.981104 4c4e.35fc.fcd8 00be.75f6.1dae 0x0800 Length: 66
    192.168.0.100.47741 > 10.10.1.100.80: S [tcp sum ok] 3809380540:3809380540(0) win 8192 <mss 1460,
3: 13:57:36.981776 00be.75f6.1dae</pre>
```

a023.9f92.2a4d

```
0x0800 Length: 66
		10.10.1.100.80 > 192.168.0.100.47741: S [tcp sum ok] 1304153587:1304153587(0) ack 3809380541 win
	4: 13:57:36.982126
a023.9f92.2a4d
00be.75f6.1dae 0x0800 Length: 54
		192.168.0.100.47741 > 10.10.1.100.80:
R
[tcp sum ok] 3809380541:3809380541(0) ack 1304153588 win 8192 (ttl 255, id 48501)
...
```

Cas 5. Transfert TCP lent (scénario 1)

Description du problème :

Le transfert SFTP entre les hôtes 10.11.4.171 et 10.77.19.11 est lent. Bien que la bande passante minimale entre les deux hôtes soit de 100 Mbits/s, la vitesse de transfert ne dépasse pas 5 Mbits/s.

Dans le même temps, la vitesse de transfert entre les hôtes 10.11.2.124 et 172.25.18.134 est nettement supérieure.

Théorie de fond :

La vitesse de transfert maximale pour un flux TCP unique est déterminée par le produit BDP (Bandwidth Delay Product). La formule utilisée est illustrée dans l'image :

May Single TCP Flow Throughput [hps] -	TCP Window (Bytes)	v 8 [hits/Buto]
wax single for flow filloughput [bps] = -	RTT (Seconds)	x o [bits/ byte]

Pour plus de détails sur le BDP, consultez les ressources ici :

- Pourquoi votre application utilise-t-elle uniquement 10 Mbit/s ? Même si la liaison est de 1 Gbit/s ?
- BRKSEC-3021 Avancé Optimisation des performances du pare-feu

Scénario 1. Transfert lent

Cette image présente la topologie :



Flux affecté :

IP source : 10.11.4.171

Adresse IP d'expédition : 10.77.19.11

Protocole : SFTP (FTP sur SSH)

Analyse de capture

Activer les captures sur le moteur FTD LINA :

<#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

Avertissement : les captures LINA sur FP1xxx et FP21xx affectent le taux de transfert du trafic qui passe par le FTD. N'activez pas les captures LINA sur les plates-formes FP1xxx et FP21xxx lorsque vous dépannez des problèmes de performances (transfert lent via le FTD). Utilisez plutôt SPAN ou un périphérique HW Tap en plus des captures sur les hôtes source et de destination. Le problème est documenté dans l'ID de bogue Cisco <u>CSCvo30697</u>.

<#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.

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Calcul du temps de parcours aller-retour (RTT)

N	lo.	Time	Source	Destination	Protocol	Length	Window size v	value 1
	- 1 2 3 4 5	0.000000 0.072521 0.000168 0.077068 0.000152	10.11.4.171 10.77.19.11 10.11.4.171 10.77.19.11 10.11.4.171	Mark/Unmark Packet Ignore/Unignore Pack Set/Unset Time Refere Time Shift	et ence	70 70 58 80 58		49640 49680 49680 49680 49680 49680
	6 7 8	0.000244 0.071545 0.000153	10.11.4.171 10.77.19.11 10.11.4.171	Packet Comment Edit Resolved Name		80 58 538		49680 (49680 2 49680 (
	9 10 11 12	0.041288 0.000168 0.030165 0.000168	10.77.19.11 10.11.4.171 10.77.19.11 10.11.4.171	Apply as Filter Prepare a Filter Conversation Filter Colorize Conversation	, ,	738 58 58 82		49680 9 49680 9 49680 9 49680 9
<		1. 70 h d	1	SCTP	,	·		
	> Ethe > 802	ne 1: 70 byt ernet II, Sr .1Q Virtual	es on wire (560 c: Cisco_f8:19:f LAN, PRI: 0, DEI	Follow Copy	,		P Stream DP Stream SL Stream	00:5d:73
2	> Inte	ernet Protoc	ol Version 4, Sr	Protocol Preferences	,	H	TTP Stream	93 Len:

Tout d'abord, identifiez le flux de transfert et suivez-le :

Modifiez la vue Wireshark pour afficher les secondes écoulées depuis le paquet affiché précédent. Ceci facilite le calcul de la RTT :

File	Edit	Vie	w Go Capture An	alyze Statistics Telepho	ny Wirele	s Tools	Help						
		~	Main Toolbar		0. 0.	् 🎹							
📕 Ар	ply a d	\sim	Filter Toolbar										
No.	Ti	~	Status Bar		Protoco	Length	Window size value	Info					
Г	10		Full Screen	F11	TCP	70	49	540 39744 →	22 [SYN]	Seq=1737026093			
	2 0		De alvas Lint		TCP	70	49	580 22 → 39	744 [SYN,	ACK] Seq=835172			
	30.		Packet List		TCP	58	49	580 39744 →	22 [ACK]	Seq=1737026094 /			
	40.	. ~	Packet Details		SSHv2	80	49	580 Server:	Protocol	(SSH-2.0-Sun_SSI			
	50.	· ~	Packet Bytes		TCP	58	49	580 39744 →	22 [ACK]	Seq=1737026094 /			
	60.	•	Time Display Format		 Da 	Date and Time of Day (1970-01-01 01:02:03.123456)							
	70.	•	Name Resolution		 Yea 	Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)							
	80.	•	Zoom		, Tin	Time of Day (01:02:03.123456) Ctrl+Alt+2							
	90.	•	20011		Se	onds Sinc	e 1970-01-01			Ctrl+Alt+3			
	10 0.	•	Expand Subtrees	Shift+Right	Se	Seconds Since Beginning of Capture Ctrl+Alt+4							
	12.0		Collapse Subtrees	Shift+Left	Se	onds Sinc	e Previous Captured	Packet		Ctrl+Alt+5			
<	12 0		Expand All	Ctrl+Right	• See	Seconds Since Previous Capital de Packet							

Le RTT peut être calculé par addition des valeurs de temps entre 2 échanges de paquets (un vers la source et un vers la destination). Dans ce cas, le paquet #2 affiche le RTT entre le pare-feu et le périphérique qui a envoyé le paquet SYN/ACK (serveur). Le paquet #3 indique le délai entre le pare-feu et le périphérique qui a envoyé le paquet ACK (client). L'ajout des 2 numéros donne une bonne estimation de la valeur de bout en bout de la valeur de transfert de l'appel :

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

RTT ≈ 80 ms

Calcul de la taille de fenêtre TCP

Développez un paquet TCP, développez l'en-tête TCP, sélectionnez Calculated window size et sélectionnez Apply as Column :

~	Transmission Control Protocol, Src	Port: 22,	Dst	Port:	39744,	Seq:	835184024,	Ack:	1758069308,	Len:	32
	Source Port: 22										
	Destination Port: 39744										
	[Stream index: 0]										
	[TCP Segment Len: 32]										
	Sequence number: 835184024										
	[Next sequence number: 835184056]									
	Acknowledgment number: 175806930	8									
	0101 = Header Length: 20 by	tes (5)									
	> Flags: 0x018 (PSH, ACK)										
	Window size value: 49680										
	[Calculated window size: 49680]										
	[Window size scaling factor: :	Expand Su	btrees	5							
	Checksum: 0x2b49 [unverified]	Collapse S	ubtree	es							
	[Checksum Status: Unverified]	Expand Al	1								
	Ungant naintan. A	Collapse A	di l			ł					
0	The scaled window size (if scaling has been										
	Mindow - [Celcul	Apply as C	olum	n							

Vérifiez la colonne Valeur de taille de fenêtre calculée pour voir quelle était la valeur de taille de fenêtre maximale pendant la session TCP. Vous pouvez également sélectionner le nom de la colonne et trier les valeurs.

Si vous testez un téléchargement de fichier (serveur > client), vous devez vérifier les valeurs annoncées par le serveur. La valeur de taille de fenêtre maximale annoncée par le serveur détermine la vitesse de transfert maximale atteinte.

Dans ce cas, la taille de la fenêtre TCP est de ≈ 50000 octets

Apply Apply	a display filter	<ctrl-></ctrl->					
No.	Time	Source	Destination	Protocol	Length	Calculated window size	Info
24	0.000091	10.11.4.171	10.77.19.11	TCP	58	49	9680 39744 → 22 [ACK] Seq=1758069341 Ack=83
24	0.000077	10.77.19.11	10.11.4.171	TCP	58	49	9680 22 → 39744 [FIN, ACK] Seq=835184152 Ac
24	0.071605	10.77.19.11	10.11.4.171	TCP	58	49	9680 22 → 39744 [ACK] Seq=835184152 Ack=175
24	0.000153	10.11.4.171	10.77.19.11	TCP	58	49	9680 39744 → 22 [FIN, ACK] Seq=1758069340 A
24	0.000443	10.11.4.171	10.77.19.11	SSHv2	90	49	9680 Client: Encrypted packet (len=32)
24	0.071666	10.77.19.11	10.11.4.171	SSHv2	154	49	9680 Server: Encrypted packet (len=96)
24	0.044050	10.11.4.171	10.77.19.11	TCP	58	49	9680 39744 → 22 [ACK] Seq=1758069308 Ack=83
24	0.073605	10.77.19.11	10.11.4.171	SSHv2	90	49	9680 Server: Encrypted packet (len=32)
24	0.000747	10.11.4.171	10.77.19.11	SSHv2	90	49	9680 Client: Encrypted packet (len=32)

Sur la base de ces valeurs et en utilisant la formule Bandwidth Delay Product, vous obtenez la bande passante théorique maximale qui peut être atteinte dans les conditions suivantes :

50000*8/0,08 = 5 Mbits/s de bande passante théorique maximale.

Cela correspond à ce que le client ressent dans ce cas.

Vérifiez de près la connexion TCP en trois étapes. Les deux côtés, et plus important encore le serveur, annoncent une valeur d'échelle de fenêtre de 0, ce qui signifie 2^0 = 1 (aucune échelle de fenêtre). Cela affecte négativement le taux de transfert :



À ce stade, il est nécessaire de prendre une capture sur le serveur, de confirmer que c'est celui qui annonce l'échelle de fenêtre = 0 et de la reconfigurer (consultez la documentation du serveur pour savoir comment faire).

Scénario 2. Transfert rapide

Examinons maintenant le bon scénario (transfert rapide via le même réseau) :

Topologie:



Le flux d'intérêt :

IP source : 10.11.2.124

Adresse IP de destination : 172.25.18.134

Protocole : SFTP (FTP sur SSH)

Activer les captures sur le moteur LINA FTD

<#root>

firepower#

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

firepower#

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

Calcul du temps de parcours aller-retour (RTT) : dans ce cas, le RTT est \approx 300 ms.

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

Calcul de la taille de fenêtre TCP : le serveur annonce un facteur d'échelle de fenêtre TCP de 7.



La taille de la fenêtre TCP du serveur est de ≈ 1600000 octets :

📕 Apply	a display filter	<ctrl-></ctrl->						
No.	Time	Source	Destination	Protocol	Length	Window size value	Calculated window size	Info
23	0.002579	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [FIN, ACK]
23	0.266847	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=
23	0.268089	172.25.18.134	10.11.2.124	SSHv2	198	12854	1645312	Server: Encrypted pack
23	0.000076	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pack
23	0.000351	172.25.18.134	10.11.2.124	SSHv2	118	12854	1645312	Server: Encrypted pack
23	0.000092	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=0
23	0.000015	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=0
23	0.000091	172.25.18.134	10.11.2.124	TCP	70	12854	1645312	22 → 57093 [ACK] Seq=6

Sur la base de ces valeurs, la formule de produit Délai de bande passante donne :

1600000*8/0,3 = vitesse de transfert théorique maximale de 43 Mbits/s

Cas 6 . Transfert TCP lent (scénario 2)

Description du problème : le transfert de fichiers FTP (téléchargement) via le pare-feu est lent.

Cette image présente la topologie :



Flux affecté :

Adresse IP source : 192.168.2.220

Adresse IP d'expédition : 192.168.1.220

Protocole : FTP

Analyse de capture

Activez les captures sur le moteur FTD LINA.

<#root>

firepower#

capture CAPI type raw-data buffer 33554432 interface INSIDE match tcp host 192.168.2.220 host 192.168.1 firepower#

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

INSIDE (CAPI) :

75 0.000412	192.168.2.220	192.168.1.220	ТСР	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	тср	Packet Comment	q=1884231612 Ack=2670020879
82 0.000091	192.168.2.220	192.168.1.220	TCP	Edit Resolved Name	4494 → 2388 [ACK] Seq=188423
83 0.000015	192.168.2.220	192.168.1.220	TCP	Edit Resolved Name	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	тср	SCTP	, 38 → 54494 [ACK] Seq=2670026
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	- cheff	e15mb)

Le contenu du flux FTP-DATA :

1	26 0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
1 3	28 1.026564	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 + 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
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 TSval=4264384 TSecr=3577288500
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=4264384
	34 0.001617	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
1	35 0.000351	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
	36 0.000458	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
	37 0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	38 0.000198	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=2669992175 SRE=2669993423
1	39 0.000077	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669990927 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=2669992175 SRE=2669994671
4	40 0.309096	192.168.1.220	192.168.2.220	тср	1314 [TCP Out-Of-Order] 2388 + 54494 [ACK] Seq=2669990927 Ack=1884231612 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
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=4264415
4	42 0.000489	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
1 4	43 0.000845	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
4	44 0.000077	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
4	45 0.000244	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
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=4264415 SLE=2669997167 SRE=2669999663
4	47 0.000504	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
1	48 0.000259	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=1884231612 Ack=2669995919 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2669997167 SRE=2670000911
1 4	49 0.918126	192.168.1.220	192.168.2.220	тср	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2669995919 Ack=1884231612 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
	50 0.000900	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670000911 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
	51 0.000519	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	52 0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	53 0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
	54 0.000015	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	55 0.000199	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670002159 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
	56 0.000229	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=1884231612 Ack=2670003407 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
	57 0.000183	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	58 0.000106	192.168.2.220	192.168.1.220	TCP	78 [ICP Window Update] 54494 + 2388 [AKK] Seq=1884231012 ACK=2670005407 Win=65280 Len=0 TSva1=3577292742 TSecr=4264507 SLE=2670004655 SRE=2670007151
	59 0.000168	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 + 2388 [ACK] Seq=1884231612 ACK=2670003407 Win=68224 Len=0 TSval=3577292743 TSecr=4264507 SLE=2670004655 SRE=2670008399
(60 0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

Le contenu de capture CAPO :

31 0.000000	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
33 1.026534	192.168.2.220	192.168.1.220	TCP	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
34 1.981400	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
35 0.000610	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
38 0.001328	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
40 0.000641	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSe 7 264384
41 0.000381	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb) 🔽
42 0.000046	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
43 0.000290	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 + 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=35072 Len=0 TSval=3577291511 TSecr=4264384 SLE=2224319408 SRE=2224320656
44 0.000076	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 + 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=37888 Len=0 TSval=3577291511 TSecr=4264384 SLE=2224319408 SRE=2224321904
45 0.309005	192.168.1.220	192.168.2.220	TCP	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224318160 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291511
46 0.000580	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
47 0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
48 0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
49 0.000076	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
50 0.000290	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415
51 0.000046	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=48768 Len=0 TSval=3577291821 TSecr=4264415 SLE=2224324400 SRE=2224326896
52 0.000412	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
53 0.000351	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 + 2388 [ACK] Seq=2157030682 Ack=2224323152 Win=51584 Len=0 TSval=3577291822 TSecr=4264415 SLE=2224324400 SRE=2224328144
54 0.918019	192.168.1.220	192.168.2.220	TCP 1	1314 [TCP Out-Of-Order] 2388 + 54494 [ACK] Seq=2224323152 Ack=2157030682 Win=66048 Len=1248 TSval=4264507 TSecr=3577291822
55 0.001007	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224328144 Win=54528 Len=0 TSval=3577292741 TSecr=4264507
56 0.000457	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
57 0.000061	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
58 0.000016	192.168.1.220	192.168.2.220	FTP-DATA	1314 [TCP Previous segment not captured] FTP Data: 1248 bytes (PASV) (RETR file15mb)
59 0.000000	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
60 0.000274	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224329392 Win=57472 Len=0 TSval=3577292742 TSecr=4264507
61 0.000214	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224330640 Win=60288 Len=0 TSval=3577292742 TSecr=4264507
62 0.000122	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
63 0.000168	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224330640 Win=65280 Len=0 TSval=3577292742 TSecr=4264507 SLE=2224331888 SRE=2224334884
64 0.000107	192.168.1.220	192.168.2.220	FTP-DATA	1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)

Principaux points :

- 1. Il existe des paquets TCP hors ordre (OOO).
- 2. Il y a une retransmission TCP.
- 3. Il y a une indication de perte de paquets (paquets abandonnés).

Conseil : enregistrez les captures lorsque vous accédez à Fichier > Exporter les paquets spécifiés. Enregistrez ensuite uniquement la plage de paquets affichée

File name:	FTD_Data_only				~
Save as type:	Wireshark/tcpdump/ pca	p (*.dmp.gz;*.dmp;*.ca	ap.gz;*.cap;*.pcap.g	jz;*.pcap)	~
Compress with gzi	p				
PacketRange				1	
		Captured	Displayed		
All packets		23988	23954		
O Selected packet	t	1	1		
O Marked packets	1	0	0		
O First to last mark	ed	0	0		
ORange:		0	0		
Remove lanored	d packets	0	0		

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Identifiez l'emplacement de perte de paquets.

Dans de tels cas, vous devez effectuer des captures simultanées et utiliser la méthode « diviser et conquérir » pour identifier le ou les segments de réseau à l'origine de la perte de paquets. Du point de vue du pare-feu, il existe 3 scénarios principaux :

- 1. La perte de paquets est causée par le pare-feu lui-même.
- 2. La perte de paquets est provoquée en aval vers le périphérique pare-feu (direction du serveur vers le client).
- 3. La perte de paquets est provoquée en amont vers le périphérique pare-feu (direction du client vers le serveur).

Perte de paquets causée par le pare-feu : afin d'identifier si la perte de paquets est causée par le pare-feu, il est nécessaire de comparer la capture d'entrée à la capture de sortie. Il existe de nombreuses façons de comparer deux captures différentes. Cette section présente une méthode d'exécution de cette tâche.

Procédure de comparaison de 2 captures afin d'identifier la perte de paquets

Étape 1. Assurez-vous que les 2 captures contiennent des paquets provenant de la même fenêtre temporelle. Cela signifie qu'il ne doit pas y avoir de paquets dans une capture qui ont été capturés avant ou après l'autre capture. Pour ce faire, vous pouvez procéder de plusieurs manières :

- Vérifiez les première et dernière valeurs d'identification IP (ID) de paquet.
- Vérifiez les valeurs d'horodatage du premier et du dernier paquet.

Dans cet exemple, vous pouvez voir que les premiers paquets de chaque capture ont les mêmes valeurs d'ID IP :

No. Time Source Destination Protocol	Length Identification	lafo
- 1 2019-10-16 16:13:44.169394 192.168.2.220 192.168.1.220 TCP	7 0x0a34 (2612)	54494 + 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=128
2 2019-10-16 16:13:45.195958 192.168.2.220 192.168.1.220 TCP	74 0x0a35 (2613)	[TCP Retransmission] 54494 → 2388 [SYN] Seq=1884231611 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577289526 TSecr=0 WS=128
3 2019-10-16 16:13:47.177542 192.168.1.220 192.168.2.220 TCP	74 0x151f (5407)	2388 -> 54494 [SYN, ACK] Seq=2669989678 Ack=1884231612 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSval=4264384 TSecr=3577288500
4 2019-10-16 16:13:47.178030 192.168.2.220 192.168.1.220 TCP	66 0x0a36 (2614)	54494 + 2388 [ACK] Sen=1884231612 Ack=2669989679 Win=29312 Len=0 TSva]=3577291588 TSecr=4264384
5 2019-10-16 16:13:47.179647 192.168.1.220 192.168.2.220 TCP	1314 0x1521 (5409)	Wireshark
6 2019-10-16 16:13:47.179998 192.168.2.220 192.168.1.220 TCP	66 0x0a37 (2615)	The Selfs Many Co. Control Analysis Confering Telephone Windows Tanks Holes
7 2019-10-16 16:13:47.180456 192.168.1.220 192.168.2.220 TCP	1314 0x1523 (5411)	File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
8 2019-10-16 16:13:47.180517 192.168.1.220 192.168.2.220 TCP	1314 0x1524 (5412)	_ ▲ ■ 点 ❷ ▶) ※ ② Q + + + 雪 著 主 🛄 @ Q Q Q Ⅲ
9 2019-10-16 16:13:47.180715 192.168.2.220 192.168.1.220 TCP	78 0x0a38 (2616)	Apply a display filter <ctrl-></ctrl->
10 2019-10-16 16:13:47.180792 192.168.2.220 192.168.1.220 TCP	78 0x0a39 (2617)	No. Time Source Destination Protocol Lendt Identification Info
11 2019-10-16 16:13:47.489888 192.168.1.220 192.168.2.220 TCP	1314 0x1525 (5413)	1 2019-10-16 16:13:44.169516 192.168.2.220 192.168.1.220 TCP 7 0x0a34 (2612) 54494 + 2388 [SYN] Seg=2157
12 2019-10-16 16:13:47.490376 192.168.2.220 192.168.1.220 TCP	66 0x0a3a (2618)	2 2019-10-16 16:13:45 196050 192-168 2 220 192-168 1 220 TCP 74 0x0a35 (2613) [TCP Retransmission] 54494
13 2019-10-16 16:13:47.490865 192.168.1.220 192.168.2.220 TCP	1314 0x1526 (5414)	3 2010-10-16 16:13:47.177450 192.168.1.220 192.168.2.220 TCP 74 0x151f (5407) 2388 + 54494 (5YN, ACK) Sec
14 2019-10-16 16:13:47.490910 192.168.1.220 192.168.2.220 TCP	1314 0x1528 (5416)	4 2019-10-16 16:13:47.178060 192.168.2.220 192.168.1.220 TCP 66.0x0036 (2614) 54494 → 2388 [ACK] Sequ2152
15 2019-10-16 16:13:47.490987 192.168.1.220 192.168.2.220 TCP	1314 0x1529 (5417)	5 2019-10-16 16:13:47 179388 192 168 1 220 192 168 2 220 TCP 1314 0v1521 (5409) 2388 + 54494 [ACK] Sen=2224
16 2019-10-16 16:13:47.491231 192.168.2.220 192.168.1.220 TCP	66 0x0a3b (2619)	6 2019-10-16 16:13:47.180029 192.168.2.220 192.168.1.220 TCP 66 0x0037 (2615) 54494 + 2388 [ACK] Seq=2157
17 2019-10-16 16:13:47.491261 192.168.2.220 192.168.1.220 TCP	78 0x0a3c (2620)	7 2019-10-16 16:13:47 180410 192-168 1-220 192-168 2-220 TCP 1314 0x1523 (5411) [TCP Previous sement not d
18 2019-10-16 16:13:47.491765 192.168.1.220 192.168.2.220 TCP	1314 0x152a (5418)	8 2019-10-16 16:13:47.180456 192.168.1.220 192.168.2.220 TCP 1314 0x1524 (5412) 2388 + 54494 [ACK] Seg=2224
19 2019-10-16 16:13:47.492024 192.168.2.220 192.168.1.220 TCP	78 0x0a3d (2621)	9 2019-10-16 16:13:47.180746 192.168.2.220 192.168.1.220 TCP 78 0x0a38 (2616) [TCP Window Undate] 54494 -
20 2019-10-16 16:13:48.410150 192.168.1.220 192.168.2.220 TCP	1314 0x152e (5422)	10 2019-10-16 16:13:47 180822 192 168 2 220 192 168 1 220 TCP 78 0x0a39 (2617) [TCP Window Undate] 54494
21 2019-10-16 16:13:48.411050 192.168.2.220 192.168.1.220 TCP	66 0x0a3e (2622)	11 2019-10-16 16:13:47.489827 192.168.1.220 192.168.2.220 TCP 1314 0x1525 (5413) [TCP 0ut-0f-0rder] 2388 + 9
22 2019-10-16 16:13:48.411569 192.168.1.220 192.168.2.220 TCP	1314 0x152f (5423)	12 2019-10-16 16:13:47.490407 192.168.2.220 192.168.1.220 TCP 66 0x0a3a (2618) 54494 + 2388 [ACK] Seg=2157
23 2019-10-16 16:13:48.411630 192.168.1.220 192.168.2.220 TCP	1314 0x1530 (5424)	13 2019-10-16 16:13:47.490819 192.168.1.220 192.168.2.220 TCP 1314 0x1526 (5414) 2388 + 54494 [ACK] Seg=2224
24 2019-10-16 16:13:48.411645 192.168.1.220 192.168.2.220 TCP	1314 0x1532 (5426)	14 2019-10-16 16:13:47.490880 192.168.1.220 192.168.2.220 TCP 1314 0x1528 (5416) [TCP Previous segment not of
25 2019-10-16 16:13:48.411660 192.168.1.220 192.168.2.220 TCP	1314 0x1533 (5427)	15 2019-10-16 16:13:47,490956 192,168,1,220 192,168,2,220 TCP 1314 0x1529 (5417) 2388 + 54494 [ACK] Seg=2224
26 2019-10-16 16:13:48.411859 192.168.2.220 192.168.1.220 TCP	66 0x0a3f (2623)	16 2019-10-16 16:13:47.491246 192.168.2.220 192.168.1.220 TCP 66 0x0a3b (2619) 54494 + 2388 [ACK] Seg=2152
27 2019-10-16 16:13:48.412088 192.168.2.220 192.168.1.220 TCP	66 0x0a40 (2624)	17 2019-10-16 16:13:47.491292 192.168.2.220 192.168.1.220 TCP 78 0x0a3c (2620) [TCP Window Update] 54494
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)		18 2019-10-16 16:13:47.491704 192.168.1.220 192.168.2.220 TCP 1314 0x152a (5418) 2388 + 54494 [ACK] Seg=2224
Ethernet II, Src: Vmware 0b:e3:cb (00:0c:29:0b:e3:cb), Dst: Cisco 9d:89:97	(50:3d:e5:9d:89:97)	19 2019-10-16 16:13:47.492055 192.168.2.220 192.168.1.220 TCP 78 0x0a3d (2621) [TCP Window Update] 54494
> Internet Protocol Version 4, Src: 192.168.2.220, Dst: 192.168.1.220	,,	28 2019-10-16 16:13:48.410074 192.168.1.220 192.168.2.220 TCP 1314 0x152e (5422) [TCP Out-Of-Orden] 2388 + 5
> Transmission Control Protocol, Src Port: 54494, Dst Port: 2388, Seg: 18842	1611, Len: 0	21 2019-10-16 16:13:48,411081 192.168.2.220 192.168.1.220 TCP 66 0x0a3e (2622) 54494 + 2388 [ACK] Seq=2157
		22 2019-10-16 16:13:48,411538 192,168,1,220 192,168,2,220 TCP 1314 0x152f (5423) 2388 → 54494 [ACK] Seg=2224
		23 2019-10-16 16:13:48.411599 192.168.1.220 192.168.2.220 TCP 1314 0x1530 (5424) 2388 + 54494 [ACK] Seq=2224

Dans le cas où ils ne sont pas les mêmes alors :

- 1. Comparez les horodatages du premier paquet de chaque capture.
- 2. À partir de la capture avec le dernier Timestamp, obtenez un filtre à partir de celui-ci, changez le filtre Timestamp de == à >= (le premier paquet) et <= (le dernier paquet), par exemple :

No.	Time	Source	Destination	Protocol	Length	Info		
4	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 → 38	3400	[S
	3 2019-10-16 16:13:43.245867	192.168.2.220	192.168.1.220	TCP	66	38400 -	21	[A
<								
∽ Fr	ame 2: 74 bytes on wire (592 bits),	74 bytes capt	ured (592 bits)					
	Encapsulation type: Ethernet (1)							
	Arrival Time: Oct 16, 2019 16:13:4	3.245638000	-A1 C D12-64	*****				
	[Time shift for this packet: 0.000	000000 sec	Expand Subtrees					
	Epoch Time: 1571235223.245638000 s	econds	Collapse Subtrees					
	[Time delta from previous captured	frame: 0.(Expand All					
	[Time delta from previous displaye	d frame: 0.	Collapse All					
	[Time since reference or first frame Number: 2	me: 0.00094	Apply as Column					
	Frame Length: 74 bytes (592 bits)		Apply as Filter	•				
	Capture Length: 74 bytes (592 bits)	Prepare a Filter	► Set	elected		l)	

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

3. Exportez les paquets spécifiés vers une nouvelle capture, sélectionnez Fichier > Exporter les paquets spécifiés et enregistrez les paquets affichés. À ce stade, les deux captures doivent contenir des paquets qui couvrent la même période. Vous pouvez maintenant commencer la comparaison des 2 captures.

Étape 2. Spécifiez le champ de paquet utilisé pour la comparaison entre les 2 captures. Exemple de champs qui peuvent être utilisés :

- Identification IP
- Numéro de séquence RTP
- Numéro de séquence ICMP

Créez une version textuelle de chaque capture contenant le champ de chaque paquet que vous avez spécifié à l'étape 1. Pour ce faire, ne laissez que la colonne d'intérêt, par exemple, si vous voulez comparer des paquets basés sur l'identification IP puis modifier la capture comme indiqué dans l'image.

-	-			Dentes 1	lineth tot		_
Time 2 2019-10-16 16:13:43.2 3 2019-10-16 16:13:43.2 4 2019-10-16 16:13:43.5	245638 192. 245867 192. 558259 192.	Ce I .168.1.220 1 .168.2.220 1 .168.1.220 1	Destination 192.168.2.220 192.168.1.220 192.168.2.220	TCP TCP FTP	Length Into 74 21 → 38400 [SYN, A(66 38400 → 21 [ACK] Se 229 Response: 220-File	Align Left Align Center Align Right	
5 2019-10-16 16:13:43.5	558274 192. 102	.168.1.220 1	92.168.2.220	TCP	126 [TCP Out-Of-Order]	Column Preferenc	:es
Wireshark · Preference	es					?	>
Appearance					_		
Columns	Displayed	Title			Type	Fields	-
Font and Colors		No.			Number		
Lavout		Time			Time (format as specified)	
Canture		Source			Source address		
Event	- T	Destination			Destination address		
Eller Puttone	- H	Protocol			Protocol		
Filter Buttons	H	Length			Packet length (buter)		
Name Resolution	H	Conyon Conyonco numb			Curtom	ten can	
Protocols	H	Sequence numb	411		Custom	ucposed	
Statistics		Source Port			Custom	udp.srcport	
Advanced		Destination Port			Custom	udp.dstport	
		ID			Custom	vlan.id	
		Fragment Offset			Custom	dtis handsha	(
		Identification			Custom	ip.id	
		More tragments			Custom	ip.flags.mf	
		Don't fragment			Custom	ip.flags.df	v
	<					>	
>	+ -						
<i>F</i>							

Le résultat :

Identification
0x150e (5390)
0xfdb0 (64944)
0x1512 (5394)
0x1510 (5392)
0xfdb1 (64945)
0xfdb2 (64946)
0xfdb3 (64947)
0x1513 (5395)
0xfdb4 (64948)
0xfdb5 (64949)
0x1516 (5398)
0x1515 (5397)
0xfdb6 (64950)
0x1517 (5399)
0xfdb7 (64951)
0x1518 (5400)
0xfdb8 (64952)
0xtdb9 (64953)
0x151b (5403)
0x151a (5402)
0xfdba (64954)
0x151c (5404)
0xtdbb (64955)
0x1510 (5405)
0X0334 (2012)
0x1abc (04950)
0x0355 (2013)
0+0-36 (3614)
Frame 23988: 66 bytes on wire (528 bits), 66 bytes cantured (528 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Oct 16, 2019 16:20:21,785130000 Central European Davlight Time
the set of

Étape 3. Créez une version textuelle de la capture (Fichier > Exporter les dissections de paquets > En tant que texte brut...), comme illustré dans l'image :

4 W	/ireshar	rk										
File	Edit	View	Go	Capture	Analyze	Stat	istics	Telephony	W	ireless	Tools	Help
	Open Open Recent Merge Import from Hex Dump Close			Ctrl+O , Ctrl+W Ctrl+S		*	• 其 📄	Q	୍ର୍	A II		
	Save				Ctrl+S							
	Save A	S			Ctrl+Shift+S							
	File Set	t				•						
	Export	Specifi	ed Pa	ckets								
	Export	Packet	Disse	ctions		•	A	s Plain Text.				
	Export	Packet	Bytes		Ctrl+Shift-	٠X	A	s CSV				
	Export	PDUs to	o File				A	s "C" Arrays				

Désactivez les options Inclure les en-têtes de colonne et Détails du paquet pour exporter uniquement les valeurs du champ affiché, comme illustré dans l'image :

Packet Range			Packet Format
	Captured	Displayed	Packet summary line
 All packets 	16514	16514	Include column beadings
 Selected packet 	1	1	Packet details:
 Marked packets 	0	0	As displayed
 First to last marked 	0	0	As displayed 🗸
O Range:	0	0	Packet Bytes
Remove Ignored packets	0	0	Each packet on a new page

Étape 4. Trier les paquets dans les fichiers. Pour ce faire, vous pouvez utiliser la commande sort de Linux :

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

Étape 5. Utilisez un outil de comparaison de texte (par exemple, WinMerge) ou la commande Linux diff pour trouver les différences entre les 2 captures.

0x0a3d	(2621)				0x0a3d	(2621)	
0x0a3e	(2622)				0x0a3e	(2622)	
0x0a3f	(2623)				0x0a3f	(2623)	
0x0a40	(2624)				0x0a40	(2624)	
0x0a41	(2625)				0x0a41	(2625)	
0x0a42	(2626)	WinMerge	e	×	0x0a42	(2626)	
0x0a43	(2627)				0x0a43	(2627)	
0x0a44	(2628)		The selected files are identical.		0x0a44	(2628)	
0x0a45	(2629)				0x0a45	(2629)	
0x0a46	(2630)	<u> </u>	Don't display this message ag	ain.	0x0a46	(2630)	
0x0a47	(2631)				0x0a47	(2631)	
0x0a48	(2632)		Ok		0x0a48	(2632)	
0x0a49	(2633)		<u></u> ~		0x0a49	(2633)	
0x0a4a	(2634)				0x0a4a	(2634)	
0x0a4b	(2635)				0x0a4b	(2635)	
0x0a4c	(2636)				0x0a4c	(2636)	
0x0a4d	(2637)				0x0a4d	(2637)	
0x0a4e	(2638)				0x0a4e	(2638)	
0v0-4f	126301				0.00-1 F	126301	
<				>	<		
.n: 27 Col:	14/14 Ch: 14/14		1252	Win	Ln: 23955	Col: 1/1 Ch: 1/1	1252

Dans ce cas, les captures CAPI et CAPO pour le trafic de données FTP sont identiques. Cela prouve que la perte de paquets n'a pas été causée par le pare-feu.

Identifiez la perte de paquets en amont/en aval.

No.	Time	Source	Destination	Protocol	Length Info
+	1 2019-10-16 16:13:44.169516	192.168.2.220	192.168.1.220	TCP	74 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3577288500 TSecr=0 WS=1
				ТСР 🥂	74 [TCP Retransmission] 54494 → 2388 [SYN] Seq=2157030681 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=35
	3 2019-10-16 16:13:47.177450	192.168.1.220	192.168.2.220	TCP	74 2388 → 54494 [SYN, ACK] Seq=2224316911 Ack=2157030682 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1 TSv
	4 2019-10-16 16:13:47.178060	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224316912 Win=29312 Len=0 TSval=3577291508 TSecr=4264384
	5 2019-10-16 16:13:47.179388	192.168.1.220	192.168.2.220	TCP	1314 2388 → 54494 [ACK] Seq=2224316912 Ack=2157030682 Win=66048 Len=1248 TSval=4264384 TSecr=3577291508
	6 2019-10-16 16:13:47.180029	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=32128 Len=0 TSval=3577291510 TSecr=4264384
	7 2019-10-16 16:13:47.180410	192.168.1.220	192.168.2.220	тср 2	1314 [TCP Previous segment not captured] 2388 \rightarrow 54494 [ACK] Seq=2224319408 Ack=2157030682 Win=66048 Len=1
	8 2019-10-16 16:13:47.180456	192.168.1.220	192.168.2.220	TCP	1314 2388 → 54494 [ACK] Seq=2224320656 Ack=2157030682 Win=66048 Len=1248 TSval=4264384 TSecr=3577291510
	9 2019-10-16 16:13:47.180746	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=35072 Len=0 TSval=357729151
	10 2019-10-16 16:13:47.180822	192.168.2.220	192.168.1.220	TCP	78 [TCP Window Update] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224318160 Win=37888 Len=0 TSval=357729151
	11 2019-10-16 16:13:47.489827	192.168.1.220	192.168.2.220	тср	1314 [TCP Out-Of-Order] 2388 → 54494 [ACK] Seq=2224318160 Ack=2157030682 Win=66048 Len=1248 TSval=4264415
	12 2019-10-16 16:13:47.490407	192.168.2.220	192.168.1.220	TCP	66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224321904 Win=40832 Len=0 TSval=3577291820 TSecr=4264415
	13 2019-10-16 16:13:47.490819	192.168.1.220	192.168.2.220	TCP	1314 2388 → 54494 [ACK] Seq=2224321904 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
	14 2019-10-16 16:13:47.490880	192.168.1.220	192.168.2.220	тср 💋	1314 [TCP Previous segment not captured] 2388 → 54494 [ACK] Seq=2224324400 Ack=2157030682 Win=66048 Len=1
	15 2019-10-16 16:13:47.490956	192.168.1.220	192.168.2.220	TCP	1314 2388 → 54494 [ACK] Seq=2224325648 Ack=2157030682 Win=66048 Len=1248 TSval=4264415 TSecr=3577291820
	16 2019-10-16 16:13:47.491246	192,168,2,220	192,168,1,220	TCP	66 54494 + 2388 [ACK] Sec=2157030682 Ack=2224323152 Win=43776 Len=0 TSval=3577291821 TSecr=4264415

Principaux points :

1. Ce paquet est une retransmission TCP. Plus précisément, il s'agit d'un paquet SYN TCP envoyé du client au serveur pour les données FTP en mode passif. Puisque le client renvoie le paquet et que vous pouvez voir le SYN initial (paquet #1), le paquet a été perdu en amont du pare-feu.



Dans ce cas, il est possible que le paquet SYN soit arrivé au serveur, mais le paquet SYN/ACK a été perdu sur le chemin du retour :



2. Un paquet provenant du serveur et Wireshark a identifié que le segment précédent n'a pas été vu/capturé. Puisque le paquet non capturé a été envoyé du serveur au client et n'a pas été vu dans la capture du pare-feu, cela signifie que le paquet a été perdu entre le serveur et le pare-feu.



Cela indique une perte de paquets entre le serveur FTP et le pare-feu.

Action 2. Effectuez des captures supplémentaires.

Effectuez des captures supplémentaires avec les captures aux points d'extrémité. Essayez d'appliquer la méthode « diviser et conquérir » pour isoler plus précisément le segment problématique à l'origine de la perte de paquets.

_				
Ν	io. Time	Source	Destination	Protocol Length Info
	155 2019-10-16 16:13:51.749845	192.168.1.220	192.168.2.220	FTP-DA. 1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	156 2019-10-16 16:13:51.749860	192.168.1.220	192.168.2.220	FTP-DA 1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	157 2019-10-16 16:13:51.749872	192.168.1.220	192.168.2.220	FTP-DA 1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	158 2019-10-16 16:13:51.750722	192.168.2.220	192.168.1.220	TCP 66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224385552 Win=180480 Len=0 TSv
	159 2019-10-16 16:13:51.750744	192.168.1.220	192.168.2.220	FTP-DA 1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	160 2019-10-16 16:13:51.750768	192.168.2.220	192.168.1.220	TCP 66 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800 Win=183424 Len=0 TSv
	161 2019-10-16 16:13:51.750782	192.168.1.220	192.168.2.220	FTP-DA., 1314 FTD Poter 1240 bytes (PASV) (RETR file15mb)
	162 2019-10-16 16:13:51.751001	192.168.2.220	192.168.1.220	TCP7 [TCP Dup ACK 160#1] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
П	163 2019-10-16 16:13:51.751024	192.168.1.220	192.168.2.220	FTP-DA 31 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	164 2019-10-16 16:13:51.751378	192.168.2.220	192.168.1.220	TCP 7. [TCP Dup ACK 160#2] 54494 → 2388 [ACK] Seq=2157050682 Ack=2224386800
	165 2019-10-16 16:13:51.751402	192.168.1.220	192.168.2.220	FTP-DA… 1314 FTP Data: 1248 bytes (PASV) (RETR file15mb)
	166 2019-10-16 16:13:51.751622	192.168.2.220	192.168.1.220	TCP 74 [TCP Dup ACK 160#3] 54494 → 2388 [ACK] Seq=2157030682 Ack=2224386800
	167 2019-10-16 16:13:51.751648	192.168.1.220	192.168.2.220	FTP-DA 231 [TCP Fast Retransmission] TP Data: 1248 bytes (PASV) (RETR file15mb)
<				
,	Ename 167: 1314 bytes on wire (10512	hits) 1314 bytes ca	ntured (10512 hits) o	n interface 0
1	Ethernet II Src: Vmuare 30:26:78 (0	0.0c.20.20.20.72) De	t: Cisco 0d:80.0h (50	· 1d+a5+0d+90+0h)
	Internet Protocol Version 4 Spc: 10	2 168 1 220 Det: 102	169 2 220	.50.65.50.65.50)
1	Transmission Control Protocol Sec B	ont: 2388 Det Port	100.2.220 100. Sec. 222438680	Ack: 2157030692 Len: 1249
1	ETD Data (1249 bytes data)	011. 2000, 051 POL		Ack. 2157050082, Een. 1246
	[Sotup from 22]			-
	[Setup method: DASV]			
	[Setup method: PASV]			
	[command: KEIK fileismo]			
	[Current working directory (]			
	[current working directory: /]			
1	Line-based text data (1 lines)			

Principaux points :

 Le récepteur (le client FTP dans ce cas) effectue le suivi des numéros de séquence TCP entrants. S'il détecte qu'un paquet a été manqué (un numéro de séquence attendu a été ignoré), il génère un paquet ACK avec ACK='numéro de séquence attendu qui a été ignoré'. Dans cet exemple, Ack=2224386800.

2. Le Dup ACK déclenche une retransmission TCP rapide (retransmission dans les 20 ms suivant la réception d'un double ACK).

Que signifient les ACK dupliqués ?

- Quelques ACK dupliqués, mais aucune retransmission réelle, indiquent qu'il y a plus de chances que des paquets arrivent dans le désordre.
- Des ACK dupliqués suivis de retransmissions réelles indiquent une certaine perte de paquets.

Action 3. Calculez le temps de traitement du pare-feu pour les paquets en transit.

Appliquez la même capture sur 2 interfaces différentes :

<#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

Exporter la capture Vérifier la différence de temps entre les paquets entrants et sortants

Cas 7 . Problème de connectivité TCP (corruption de paquet)

Description du problème :

Le client sans fil (192.168.21.193) tente de se connecter à un serveur de destination (192.168.14.250 - HTTP) et il existe deux scénarios différents :

- Lorsque le client se connecte au point d'accès « A », la connexion HTTP ne fonctionne pas.
- Lorsque le client se connecte au point d'accès « B », la connexion HTTP fonctionne.

Cette image présente la topologie :



Flux affecté :

Adresse IP source : 192.168.21.193

Adresse IP d'expédition : 192.168.14.250

Protocole : TCP 80

Analyse de capture

Activer les captures sur le moteur FTD LINA :

<#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

Captures - Scénario fonctionnel :

Comme base de référence, il est toujours très utile de disposer de captures à partir d'un scénario dont le fonctionnement a été vérifié.

Cette image montre la capture effectuée sur l'interface NGFW INSIDE

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

Cette image montre la capture effectuée sur l'interface EXTERNE du pare-feu de nouvelle génération.

No.	Time	Source	Destination	Protocol	Length Info
_	1 2013-08-08 17:03:25.554872	192.168.21.193	192.168.14.250	TCP	66 1055 → 80 [SYN] Seq=1839800324 Win=65535 Len=0 MSS=1380 SACK_PERM=1
	2 2013-08-08 17:03:25.555177	192.168.14.250	192.168.21.193	TCP	66 80 → 1055 [SYN, ACK] Seq=521188628 Ack=1839800325 Win=64240 Len=0 MSS=1460 SACK_PERM=1
	3 2013-08-08 17:03:25.579926	192.168.21.193	192.168.14.250	TCP	58 1055 → 80 [ACK] Seq=1839800325 Ack=521188629 Win=65535 Len=0
	4 2013-08-08 17:03:25.841112	192.168.21.193	192.168.14.250	HTTP	370 GET /ttest.html HTTP/1.1
	5 2013-08-08 17:03:25.848451	192.168.14.250	192.168.21.193	TCP	1438 80 → 1055 [ACK] Seq=521188629 Ack=1839800637 Win=63928 Len=1380 [TCP segment of a reassembled PDU
	6 2013-08-08 17:03:25.848512	192.168.14.250	192.168.21.193	HTTP	698 HTTP/1.1 404 Not Found (text/html)
	7 2013-08-08 17:03:25.858476	192.168.21.193	192.168.14.250	TCP	58 1055 → 80 [ACK] Seq=1839800637 Ack=521190649 Win=65535 Len=0
	8 2013-08-08 17:03:34.391779	192.168.21.193	192.168.14.250	HTTP	369 GET /test.html HTTP/1.1
	9 2013-08-08 17:03:34.395456	192.168.14.250	192.168.21.193	HTTP	586 HTTP/1.1 200 OK (text/html)
	10 2013-08-08 17:03:34.606368	192.168.21.193	192.168.14.250	TCP	58 1055 → 80 [ACK] Seq=1839800948 Ack=521191177 Win=65007 Len=0
	11 2013-08-08 17:03:40.739646	192.168.21.193	192.168.14.250	HTTP	483 GET /test.html HTTP/1.1
L	12 2013-08-08 17:03:40.741523	192.168.14.250	192.168.21.193	HTTP	271 HTTP/1.1 304 Not Modified

Principaux points :

- 1. Les deux captures sont presque identiques (considérez la randomisation ISN).
- 2. Il n'y a aucune indication de perte de paquets.
- 3. Pas de paquets hors d'ordre (OOO)
- 4. Il existe 3 requêtes HTTP GET. Le premier reçoit un 404 'Not Found', le second un 200 'OK' et le troisième un 304 'Not Modified' message de redirection.

Captures - Scénario d'erreur connue :

Contenu de la capture d'entrée (CAPI).

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

Principaux points :

- 1. Il existe une connexion TCP en trois étapes.
- 2. Il existe des retransmissions TCP et des indications de perte de paquets.
- 3. Il y a un paquet (TCP ACK) qui est identifié par Wireshark comme Malformed.

Cette image présente le contenu de la capture de sortie (CAPO).

-					
No.	Time	Source	Destination	Protocol	Length Info
—	1 2013-08-08 15:33:31.909514	192.168.21.193	192.168.14.250	TCP	66 3072 → 80 [SYN] Seq=230342488 Win=65535 Len=0 MSS=1380 SACK_PERM=1
	2 2013-08-08 15:33:31.909804	192.168.14.250	192.168.21.193	TCP	66 80 → 3072 [SYN, ACK] Seq=268013986 Ack=230342489 Win=64240 Len=0 MSS=1460 SACK_PERM=1
	3 2013-08-08 15:33:31.913298	192.168.21.193	192.168.14.250	ТСР	60 3072 → 80 [ACK] Seq=230342489 Ack=268013987 Win=65535 Len=2[Malformed Packet]
	4 2013-08-08 15:33:31.913633	192.168.14.250	192.168.21.193	HTTP	222 HTTP/1.1 400 Bad Request (text/html)
	5 2013-08-08 15:33:31.980357	192.168.21.193	192.168.14.250	TCP	369 [TCP Retransmission] 3072 → 80 [PSH, ACK] Seq=230342489 Ack=268013987 Win=65535 Len=311
	6 2013-08-08 15:33:32.155692	192.168.14.250		TCP 🧉	58 [TCP ACKed unseen segment] 80 → 3072 [ACK] Seq=268014152 Ack=230342800 Win=63929 Len=0
	7 2013-08-08 15:33:34.871430			тср 🤜	Z22 [TCP Retransmission] 80 → 3072 [FIN, PSH, ACK] Seq=268013987 Ack=230342800 Win=63929 Len=164
	8 2013-08-08 15:33:34.894759	192.168.21.193	192.168.14.250	TCP	60 3072 → 80 [ACK] Seq=230342800 Ack=268014152 Win=65371 Len=2
	9 2013-08-08 15:33:34.933575	192.168.21.193	192.168.14.250	TCP	60 [TCP Retransmission] 3072 → 80 [FIN, ACK] Seq=230342800 Ack=268014152 Win=65371 Len=2
	10 2013-08-08 15:33:34.933774				58 [TCP ACKed unseen segment] 80 → 3072 [ACK] Seq=268014152 Ack=230342803 Win=63927 Len=0
	11 2013-08-08 15:33:35.118524	192.168.21.193	192.168.14.250	TCP	66 3073 → 80 [SYN] Seq=2731219422 Win=65535 Len=0 MSS=1380 SACK_PERM=1
	12 2013-08-08 15:33:35.118707	192.168.14.250	192.168.21.193	TCP	66 80 → 3073 [SYN, ACK] Seq=2453407925 Ack=2731219423 Win=64240 Len=0 MSS=1460 SACK_PERM=1
	13 2013-08-08 15:33:35.121591	192.168.21.193	192.168.14.250	TCP	60 3073 → 80 [ACK] Seq=2731219423 Ack=2453407926 Win=65535 Len=2[Malformed Packet]
	14 2013-08-08 15:33:35.121652	192.168.21.193	192.168.14.250	тср	371 [TCP Out-Of-Order] 3073 → 80 [PSH, ACK] Seq=2731219423 Ack=2453407926 Win=65535 Len=313
	15 2013-08-08 15:33:35.121865	192.168.14.250	192.168.21.193	HTTP	222 HTTP/1.1 400 Bad Request (text/html)
	16 2013-08-08 15:33:35.124673	192.168.21.193	192.168.14.250	TCP	60 3073 → 80 [ACK] Seq=2731219736 Ack=2453408091 Win=65371 Len=2
	17 2013-08-08 15:33:35.124810	192.168.14.250	192.168.21.193	тср	58 [TCP ACKed unseen segment] 80 → 3073 [ACK] Seq=2453408091 Ack=2731219738 Win=63925 Len=0
	18 2013-08-08 15:33:35.126061				60 [TCP Spurious Retransmission] 3073 → 80 [FIN, ACK] Seq=2731219736 Ack=2453408091 Win=65371 Len=2
	19 2013-08-08 15:33:35.126229	192.168.14.250	192.168.21.193		58 [TCP ACKed unseen segment] 80 → 3073 [ACK] Seq=2453408091 Ack=2731219739 Win=63925 Len=0

Principaux points :

Les deux captures sont presque identiques (considérez la randomisation ISN) :

- 1. Il existe une connexion TCP en trois étapes.
- 2. Il existe des retransmissions TCP et des indications de perte de paquets.
- 3. Il y a un paquet (TCP ACK) qui est identifié par Wireshark comme Malformed.

Vérifiez le paquet mal formé :

	Time	C	Destination	Destand	Locath Tafa							
NO.	1 2012 00 00 15:22:21 000102	500rce	Destination	Protocol	Length Info	ECMUT	C 4004766000 US- 65505 L 0 MCC 4460 CACK DEDM 4					
E	1 2013-08-08 15:33:31.909193	192.108.21.193	192.108.14.200	TCP	00 30/2 → 80 ((00 - 2072)	[SYN]	Seq=4231/00828 W1n=05535 Len=0 FISS=1400 SACK_PERM=1					
Y	2 2013-08-08 15:33:31.909849	192.168.14.250	192.168.21.193	TCP	00 80 → 3072	LSYN,	ALK] SEQ=86/5/5959 ACK=4231/66829 WIN=64240 LEN=0 PISS=1380 SALK_PERPI=1					
	3 2013-08-08 15:33:31.913267	192.168.21.193	192.168.14.250	тср	60 3072 → 80	[ACK]	Seq=4231766829 Ack=867575960 Win=65535 Len=2[Malformed Packet]					
>	Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)											
>	Ethernet II, Src: BelkinIn_63:90:f3 (ec:1a:59:63:90:f3), Dst: Cisco_61:cc:9b (58:8d:09:61:cc:9b)											
>	802.1Q Virtual LAN, PRI: 0, DEI: 0), ID: 20										
>	Internet Protocol Version 4, Src:	192.168.21.193, D	t: 192.168.14.250									
~	Transmission Control Protocol, Sro	Port: 3072, Dst F	Port: 80, Seq: 4231	1766829,	Ack: 867575960, Le	en: 2	2					
	Source Port: 3072											
	Destination Port: 80											
	[Stream index: 0]											
	[TCP Segment Len: 2]											
	Sequence number: 4231766829											
	[Next sequence number: 42317668	31]										
	Acknowledgment number: 86757596	0										
	0101 = Header Length: 20 b	vtes (5)										
	> Flags: 0x010 (ACK)											
	Window size value: 65535											
	[Calculated window size: 65535]											
	[Window size scaling factor: -2	(no window scalin	g used)]									
	Checksum: 0x01bf [unverified]	(0/1									
	[Checksum Status: Unverified]											
	Urgent pointer: 0											
	> [SEO/ACK analysis]											
	> [Timestamps]											
	TCP pavload (2 bytes)	-										
~	[Malformed Packet: Tunnel Socket]	1										
	<pre> [Expert Info (Error/Malformed); </pre>	Malformed Packet	(Exception occurre	d)]								
	[Malformed Packet (Exception	occurred)]										
	[Severity level: Error]											
	[Group: Malformed]											
00	00 58 8d 09 61 cc 9b ec 1a 59 63	90 f3 81 00 00 14	X ·· a ·· ·· Yc ····									
00	10 08 00 45 00 00 2a 7f 1d 40 00	80 06 d5 a4 c0 a8	···Е··*·· @·····									
00	20 15 c1 c0 a8 0e fa 0c 00 00 50	fc 3b amad 33 b6	······ ·P <u>·;</u> ··	3.								
00	30 28 98 50 10 ff ff 01 bf 00 00	00 00 4	(.p									
		-										

Principaux points :

- 1. Le paquet est identifié comme étant un paquet malformé par Wireshark.
- 2. Sa longueur est de 2 octets.
- 3. La charge utile TCP est de 2 octets.
- 4. La charge utile est de 4 zéros supplémentaires (00 00).

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Effectuez des captures supplémentaires. Incluez des captures au niveau des points d'extrémité et, si possible, essayez d'appliquer la méthode « diviser et conquérir » pour isoler la source de la corruption du paquet, par exemple :



Dans ce cas, les 2 octets supplémentaires ont été ajoutés par le pilote d'interface « A » du commutateur et la solution était de remplacer le commutateur qui cause la corruption.

Cas 8 . Problème de connectivité UDP (paquets manquants)

Description du problème : les messages Syslog (UDP 514) ne sont pas visibles sur le serveur Syslog de destination.

Cette image présente la topologie :



Flux affecté :

Adresse IP source : 192.168.1.81

Adresse IP du poste : 10.10.1.73

Protocole : UDP 514

Analyse de capture

Activer les captures sur le moteur FTD LINA :

<#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

Les captures FTD ne montrent aucun paquet :

<#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
```

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Vérifiez la table de connexion FTD.

Pour vérifier une connexion spécifique, vous pouvez utiliser cette syntaxe :

```
<#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
```

Principaux points :

- 1. Les interfaces d'entrée et de sortie sont identiques (demi-tour).
- 2. Le nombre d'octets a une valeur importante (~5 GoYtes).
- 3. Le drapeau « o » indique le débit de déchargement (HW accéléré). C'est la raison pour laquelle les captures FTD n'affichent aucun paquet. Le déchargement de flux est uniquement pris en charge sur les plates-formes 41xx et 93xx. Dans ce cas, le périphérique est un 41xx.

Action 2. Effectuez des captures au niveau du châssis.

Connectez-vous au gestionnaire de châssis Firepower et activez la capture sur l'interface d'entrée (E1/2 dans ce cas) et les interfaces de fond de panier (E1/9 et E1/10), comme illustré dans l'image





Au bout de quelques secondes :

Capture Session Filter List					
CAPI	Drop Count: 4	0103750	Operational State: DOWN - Mem	ory_Overshoot	
Interface Name	Filter	File Size (in bytes)	File Name	Device Name	
Ethernet1/10	None	276	CAPI-ethernet-1-10-0.pcap	mzafeiro_FTD	\rightarrow
Ethernet1/9	None	132276060	CAPI-ethernet-1-9-0.pcap	mzafeiro_FTD	\pm
Ethernet1/2	None	136234072	CAPI-ethernet-1-2-0.pcap	mzafeiro_FTD	\mathbf{A}

Conseil : dans Wireshark, excluez les paquets étiquetés VN pour éliminer la duplication des paquets au niveau de l'interface physique

Avant :

4 C	API-ethernet-1	-2-0.pcap				
File	Edit View	Go Capture Analyze	Statistics Telephony	Wireless	Tools	Help
41				00	2 11	7-b
	- 12 V		* 2 * X 🖬 🔳	44	ч Ш	
Apply a display filter <ctrl-></ctrl->						
No.	Time	Source	Destination	Protocol	Length	Info
	10.0000_	Cisco_61:5a:9c	Spanning-tree-(f	STP	64	4 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	4 RST. Root = 32768/0/00:11:bc:88:08:c9 Cost = 8 Port = 0x802d
	3 0.0532	Vmware_85:4f:ca	Broadcast	ARP	70	0 Who has 192.168.103.111? Tell 192.168.103.112
	4 0.0000	Vmware_85:4f:ca	Broadcast	ARP	64	4 Who has 192.168.103.111? Tell 192.168.103.112
	5 0.5216	Vmware_85:2f:00	Broadcast	ARP	70	0 Who has 10.10.10.1? Tell 10.10.10.10
	6 0.0000	Vmware_85:2f:00	Broadcast	ARP	64	4 Who has 10.10.10.1? Tell 10.10.10.10
	7 0.5770	Vmware_85:2f:00	Broadcast	ARP	70	0 Who has 10.10.10.1? Tell 10.10.10.10
	8 0.0000	Vmware_85:2f:00	Broadcast	ARP	64	4 Who has 10.10.10.1? Tell 10.10.10.10
	9 0.8479	Cisco_61:5a:9c	Spanning-tree-(f	STP	64	4 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	4 RST. Root = 32768/0/00:11:bc:88:08:c9
	11 0.1520_	Vmware_85:2f:00	Broadcast	ARP	70	0 Who has 10.10.10.1? Tell 10.10.10.10
	12 0.0000_	Vmware_85:2f:00	Broadcast	ARP	64	4 Who has 10.10.10.1? Tell 10.10.10.10
	13 0.8606	Vmware_85:4f:ca	Broadcast	ARP	70	ð Who has 192.168.103.111? Tell 192.168.103.112
	14 0.0000	Vmware_85:4f:ca	Broadcast	ARP	64	4 Who has 192.168.103.111? Tell 192.168.103.112
	15 0.1655	192.168.0.101	173.38.200.100	DNS	91	1 Standard query 0x4a9f A 2.debian.pool.ntp.org
	16 0.0000	192.168.0.101	173.38.200.100	DNS	85	5 Standard query 0x4a9f A 2.debian.pool.ntp.org
	17 0.0000	192.168.0.101	173.38.200.100	DNS	91	1 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	1 Standard query 0x4a9f A 2.debian.pool.ntp.org
	20 0.0000	192.168.0.101	173.38.200.100	DNS	85	5 Standard query 0x4a9f A 2.debian.pool.ntp.org

Après :

4	CAPI-ethernet-1-2-0.pcap					
Eile	Edit View Go Capture	Analyze Statistics	Telephony Wireless	Iools Help		
4	I 🖉 🔘 📘 🗎 🗙 🏹	9 🗰 🔿 🖀 ∓	1 I I Q Q (a, 11		
II s	yslog && !vntag					
No.	Time	Source	Destination	Protocol	Length Ti	Time to live Info
	1334 0.000000000	192.168.1.81	10.10.1.73	Syslog	147	255 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1336 0.00078873	192.168.1.81	10.10.1.73	Syslog	147	254 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1338 0.00015099	192.168.1.81	10.10.1.73	Syslog	147	253 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1340 0.000128919	192.168.1.81	10.10.1.73	Syslog	131	255 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
	1342 0.000002839	192.168.1.81	10.10.1.73	Syslog	147	252 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1344 0.000137974	192.168.1.81	10.10.1.73	Syslog	131	254 LQCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET FIREWALL:192.168.1.71\n
	1346 0.000002758	192.168.1.81	10.10.1.73	Syslog	147	251 31.4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1348 0.000261845	192.168.1.81	10.10.1.73	Syslog	131	253 Leval4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
	1350 0.000002736	192.168.1.81	10.10.1.73	Syslog	147	250 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1352 0.000798149	192.168.1.81	10.10.1.73	Syslog	200	255 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
	1354 0.000498621	192.168.1.81	10.10.1.73	Syslog	131	252 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET FIREWALL:192.168.1.71\n
	1356 0.000002689	192.168.1.81	10.10.1.73	Syslog	147	249 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1358 0.000697783	192.168.1.81	10.10.1.73	Syslog	195	255 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
	1360 0.000599702	192.168.1.81	10.10.1.73	Syslog	151	255 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET FIREWALL:192.168.1.71
	1362 0.000002728	192.168.1.81	10.10.1.73	Syslog	200	254 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
	1364 0.000499914	192.168.1.81	10.10.1.73	Syslog	131	251 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET FIREWALL:192.168.1.71\n
	1366 0.000697761	192.168.1.81	10.10.1.73	Syslog	147	248 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1368 0.000169137	192.168.1.81	10.10.1.73	Syslog	195	254 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
	1370 0.000433196	192.168.1.81	10.10.1.73	Syslog	151	254 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
	1372 0.000498718	192.168.1.81	10.10.1.73	Syslog	200	253 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
	1374 0.000002849	192.168.1.81	10.10.1.73	Syslog	131	250 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n
	1376 0.000596345	192.168.1.81	10.10.1.73	Syslog	147	247 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host identity:192.168.1.81 dur
	1378 0.000600157	192.168.1.81	10.10.1.73	Syslog	195	253 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302021: Teardown ICMP connection for faddr 192.168.1.7
	1380 0.000002772	192.168.1.81	10.10.1.73	Syslog	151	253 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609002: Teardown local-host NET_FIREWALL:192.168.1.71
	1382 0.000600947	192.168.1.81	10.10.1.73	Syslog	200	252 LOCAL4.INFO: Oct 15 2019 07:47:17: %ASA-6-302020: Built inbound ICMP connection for faddr 192.16
	1384 0.000498808	192.168.1.81	10.10.1.73	Syslog	131	249 LOCAL4.DEBUG: Oct 15 2019 07:47:17: %ASA-7-609001: Built local-host NET_FIREWALL:192.168.1.71\n

Principaux points :

- 1. Un filtre d'affichage est appliqué pour supprimer les doublons de paquets et afficher uniquement les syslogs.
- 2. La différence entre les paquets se situe au niveau des microsecondes. Cela indique un débit de paquets très élevé.
- 3. La valeur de durée de vie (TTL) diminue continuellement. Cela indique une boucle de paquets.



Action 3. Utilisez Packet Tracer.

Comme les paquets ne traversent pas le moteur LINA du pare-feu, vous ne pouvez pas effectuer de trace en direct (capture avec trace), mais vous pouvez suivre un paquet émulé avec packet-tracer :

<#root>

firepower#

packet-tracer input INSIDE udp 10.10.1.73 514 192.168.1.81 514

Phase: 1 Type: CAPTURE Subtype: Result: ALLOW Config: Additional Information: MAC Access list Phase: 2 Type: ACCESS-LIST Subtype: Result: ALLOW Config: Implicit Rule Additional Information: MAC Access list Phase: 3 Type: FLOW-LOOKUP Subtype: Result: ALLOW Config: Additional Information: Found flow with id 25350892, using existing flow Phase: 4 Type: SNORT Subtype: Result: ALLOW Config: Additional Information: Snort Verdict: (fast-forward) fast forward this flow Phase: 5 Type: ROUTE-LOOKUP Subtype: Resolve Egress Interface Result: ALLOW Config: Additional Information: found next-hop 192.168.1.81 using egress ifc INSIDE Phase: 6 Type: ADJACENCY-LOOKUP Subtype: next-hop and adjacency Result: ALLOW Config: Additional Information: adjacency Active next-hop mac address a023.9f92.2a4d hits 1 reference 1 Phase: 7 Type: CAPTURE Subtype: Result: ALLOW Config: Additional Information: MAC Access list Result: input-interface: INSIDE input-status: up input-line-status: up

output-interface: INSIDE

output-status: up output-line-status: up Action: allow

Mesure 4. Confirmez le routage FTD.

Vérifiez la table de routage du pare-feu pour voir s'il existe des problèmes de routage :

<#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
```

Principaux points :

- 1. La route pointe vers l'interface de sortie correcte.
- 2. La route a été apprise il y a quelques minutes (0:02:37).

Action 5. Confirmez la disponibilité de la connexion.

Vérifiez la disponibilité de la connexion pour savoir quand cette connexion a été établie :

```
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 Os,
uptime 3m49s
```

, timeout 2mOs, bytes 4801148711

Point clé :

1. La connexion a été établie il y a environ 4 minutes (avant l'installation de la route EIGRP dans la table de routage)

Mesure no 6. Effacez la connexion établie.

Dans ce cas, les paquets correspondent à une connexion établie et sont routés vers une interface de sortie incorrecte ; cela provoque une boucle. Ceci est dû à l'ordre des opérations du pare-feu :

- 1. Recherche de connexion établie (priorité sur la recherche de table de routage globale).
- 2. Recherche NAT (Network Address Translation) : la phase UN-NAT (destination NAT) est prioritaire sur la recherche PBR et la recherche de route.
- 3. Routage basé sur des politiques (PBR)
- 4. Recherche de table de routage globale

Comme la connexion n'expire jamais (le client Syslog envoie continuellement des paquets pendant que le délai d'inactivité de la connexion UDP est de 2 minutes), il est nécessaire d'effacer manuellement la connexion :

<#root>

firepower#

clear conn address 10.10.1.73 address 192.168.1.81 protocol udp port 514

1 connection(s) deleted.

Vérifiez qu'une nouvelle connexion est établie :

```
<#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
```

Mesure 7. Configurez le délai de conn flottant.

C'est la solution appropriée pour résoudre le problème et éviter un routage non optimal, en particulier pour les flux UDP. Accédez à Devices > Platform Settings > Timeouts et définissez la valeur :

SMTP Server	H.323	Default v		0:05:00	(0:0:0 or 0:0:0 - 1193:0:0)
SNMP	SIP	Default		0:30:00	(0:0:0 or 0:5:0 - 1193:0:0)
SSL	SIP Media	Default •	n i	0:02:00	(0:0:0 or 0:1:0 - 1193:0:0)
Syslog			ו נ ר ר		
Timeouts	SIP Disconnect:	Default v	J	0:02:00	(0:02:0 or 0:0:1 - 0:10:0)
Time Synchronization	SIP Invite	Default 🔻		0:03:00	(0:1:0 or 0:1:0 - 0:30:0)
UCAPL/CC Compliance	SIP Provisional Media	Default •		0:02:00	(0:2:0 or 0:1:0 - 0:30:0)
	Floating Connection	Custom 🔻][0:00:30	(0:0:0 or 0:0:30 - 1193:0:0)
	Xlate-PAT	Default •		0:00:30	(0:0:30 or 0:0:30 - 0:5:0)

Vous pouvez trouver plus de détails sur le délai d'attente du conn flottant dans le Guide de référence des commandes :

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

Cas 9 . Problème de connectivité HTTPS (scénario 1)

Description du problème : la communication HTTPS entre le client 192.168.201.105 et le serveur 192.168.202.101 ne peut pas être établie

Cette image présente la topologie :



Flux affecté :

Adresse IP source : 192.168.201.111

Adresse IP d'expédition : 192.168.202.111

Protocole : TCP 443 (HTTPS)

Analyse de capture

Activer les captures sur le moteur FTD LINA :

L'adresse IP utilisée dans la capture OUTSIDE est différente en raison de la configuration de la traduction d'adresses de port.

<#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

Cette image montre la capture effectuée sur l'interface INSIDE du pare-feu de nouvelle génération .

No.	Time	Source	Destination	Protocol	Length Identification	Irfo
-	38 2018-02-01 10:39:35.187887	192.168.201.111	192.168.202.111	TCP	78 0x2f31 (12081)	6666 → 443 [SYN] Seq=2034865631 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=192658158 TSecr=0 WS=1
	39 2018-02-01 10:39:35.188909	192.168.202.111	192.168.201.111	TCP	78 0x0000 (0)	😃 443 → 6666 [SYN, ACK] Seq=4086514531 Ack=2034865632 Win=28960 Len=0 MSS=1380 SACK_PERM=1 TSval=31
	40 2018-02-01 10:39:35.189046	192.168.201.111	192.168.202.111	TCP	70 0x2f32 (12082)	6666 → 443 [ACK] Seq=2034865632 Ack=4086514532 Win=29312 Len=0 TSval=192658158 TSecr=3119615816
	41 2018-02-01 10:39:35.251695	192.168.201.111	192.168.202.111	TLSv1	326 0x2f33 (12083)	Client Hello
	42 2018-02-01 10:39:35.252352	192.168.202.111	192.168.201.111	TCP	70 0xefb4 (61364)	443 → 6666 [ACK] Seq=4086514532 Ack=2034865888 Win=8192 Len=0 TSval=3119615816 TSecr=192658174
L	43 2018-02-01 10:40:05.317320	192.168.202.111	192.168.201.111	TCP	70 0xd8c3 (55491)	443 → 6666 [RST] Seq 4086514532 Win 8192 Len 0 TSval 3119645908 TSecr 0

Principaux points :

- 1. Il existe une connexion TCP en trois étapes.
- 2. La négociation SSL commence. Le client envoie un message Hello Client.
- 3. Un accusé de réception TCP est envoyé au client.
- 4. Un RST TCP est envoyé au client.
Cette image montre la capture effectuée sur l'interface EXTERNE du pare-feu de nouvelle génération.



Principaux points :

- 1. Il existe une connexion TCP en trois étapes.
- 2. La négociation SSL commence. Le client envoie un message Hello Client.
- 3. Des retransmissions TCP sont envoyées du pare-feu vers le serveur.
- 4. Un RST TCP est envoyé au serveur.

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Effectuez des captures supplémentaires.

Une capture effectuée sur le serveur révèle que le serveur a reçu les Hello du client TLS avec la somme de contrôle TCP corrompue et les abandonne silencieusement (il n'y a pas de TCP RST ou tout autre paquet de réponse vers le client) :



Lorsque vous mettez tout ensemble :

Dans ce cas, pour comprendre, il est nécessaire d'activer sur Wireshark l'option Valider la somme de contrôle TCP si possible. Naviguez jusqu'à Edit > Preferences > Protocols > TCP, comme indiqué dans l'image.

Wireshark - Preference	s	?	×
Steam IHS D ^ STP STT STUN SUA SV SV SVNC SYNC SYNCHROPH Synergy Syslog T.38 TACACS TACACS TACACS+ TALI TAPA TCAP TCP TCPENCAP X	Transmission Control Protocol ✓ Show TCP summary in protocol tree ✓ Validate the TCP checksum if possible ✓ Allow subdissector to reassemble TCP streams ✓ Analyze TCP sequence numbers Relative sequence numbers Scaling factor to use when not available from capture ✓ Track number of bytes in flight ✓ Calculate conversation timestamps Try heuristic sub-dissectors first Ignore TCP Timestamps in summary ✓ Do not call subdissectors for error packets ✓ TCP Experimental Options with a Magic Number Display process information via IPFDX TCP UDP port		~
	OK Cancel	Help	>

Dans ce cas, il est utile de placer les captures côte à côte afin d'obtenir une image complète :

	Wireshark										- 🗆 ×	
ь	File Edit View Go Ca	apture Analyze Statis	tics Telephony W	reless Tools Help								
L	(8 8 9 4 4 4 5	T + 🗖 🗖 Q	0.0.1								
ľ	Tro stream as 1											
Ľ	a orbitation and a			Particular.		And the second second					D and - Depression	
Ľ	No. Time		Source	Destination	Protocol	Length Joenthication	20043	0	[C101] C 20240([C24.10]	- 20200	A LOS A MER AND CHEM A TOUR A TOUR ADDREAD TO TO A DIR ADD	
	38 2018-02-01 1	0:39:35.18/88/	192.168.201.111	192.108.202.111	TCP	78 0x2131 (1	2031) 00	00 + 443	[SYN] Seq=2034805031 W10	n=29200	20240656522 USe_20060 Los_0 MCC_1200 CACK DE0M_1 Tours1_2100615016 Tours_1020	
E	40.2018-02-01 1	0:39:32.188909	192.168.202.111	102.168.201.111	TCP	78 0x0000 (0	20922	5 + 0000 66 + 443	[STN, MCK] Seque0803143.	31 ACK=2	2034803032 Win=28900 Len=0 FSS=1380 SMLK_FERT=1 ISV81=3119012810 ISECT=1920	
L	41 2018-02-01 1	0.39.35.169640	192.168.201.111	192.168.202.111	TISH	326 0x2f33 (1	2002 2002	ient Hall	[MCK] SEQ=2034003032 MC	K=408031	14352 WIN=25512 CEN=0 15981=152036136 13E(1=5115013610	
L	42 2018-02-01 1	8:39:35.252352	192.168.202.111	192.168.201.111	TCP	70 0xefb4 (6	1364) 44	3 + 6666	/ [ACK] Sequ4086514532 Act	k=203486	65888 Win=8192 Len=0 TSval=3119615816 TSecr=192658174	
h	43 2018-02-01 1	0:40:05.317320	192.168.202.111	192.168.201.111	TCP	70 0xd8c3 (5	5491) 5 44	3 + 6666	RST1 Seq=4086514532 Wit	n=8192 L	Len-@ TSval=3119645988 TSecr=@	
r		01401031321320		172.1100.1201.1111		10 0100005 (5			tipil ped-incorrection int			
н												
L			FTD_CAPO_PAT	f.pcap							-	o x
L			File Edit View	Go Capture Analyze :	Statistics	Telephony Wireless	Tools Help					
L			▲■ ₫ ⊙]	🖹 🕅 🖉 🔍 🗰 🗰	留下土	. 🔚 🗐 Q, Q, G	11					
Т			tcp.stream eq 1								🛛 📼 🔹 Big	ression +
L			No. Time		Source	Des	tination	Protocol	Length Identification		2/10	
L			33 2018-0	2-01 10:39:35.188192	192.1	68.202.11 192	2.168.202.111	TCP	78 0x2f31 (12081)	-	15880 + 443 [SYN] Seq=2486930707 Win=29200 Len=0 MSS=1380 SACK_PERM=1 TSval=192	2658158 TSe
н			34 2018-0	2-01 10:39:35.188527	192.1	68.202.111 192	2.168.202.11	TCP	78 0x0000 (0)	1	443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=28960 Len=0 MSS=1460 S	ACK_PERM=1
L			35 2018-0	2-01 10:39:35.189214	192.1	68.202.11 192	2.168.202.111	TCP	70 0x2f32 (12082)	X	15880 + 443 [ACK] Seq=2486930708 Ack=3674405383 Win=29312 Len=0 T5val=192658158	3 TSecr=311
L			36 2018-0	2-01 10:39:35.252397	192.1	68.202.11 192	2.168.202.111		257 0xcd36 (52534)	2	Client Hello	
н			37 2018-0						257 0xb905 (47365)			3192 [TCP C
н			38 2018-0	2-01 10:39:41.297332	192.1	68.202.11 192	2.168.202.111		257 0x88af (34991)	4	[TCP Retransmission] 15880 → 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=4	3192 [TCP C
T			39 2018-0	2-01 10:39:49.309569	192.1	68.202.11 192	2.168.202.111		257 0xf68a (63114)		[TCP Retransmission] 15880 + 443 [PSH, ACK] Seq=2486930708 Ack=3674405383 Win=1	3192 [TCP C
T			40 2018-0	2-01 10:40:05.317305	192.1	68.202.11 192	2.168.202.111	TCP	70 0xd621 (54817)	6	15880 + 443 [RST] Seq=2486930895 Win=8192 [TCP CHECKSUM INCORRECT] Len=0 TSval	192688266
1			41 2018-0	2-01 10:40:06.790700	192.1	68.202.111 192	2.168.202.11	TCP	78 0x0000 (0)	-	[TCP Retransmission] 443 → 15880 [SYN, ACK] Seq=3674405382 Ack=2486930708 Win=2	28960 Len=0

Principaux points :

- 1. Il existe une connexion TCP en trois étapes. Les ID IP sont identiques. Cela signifie que le flux n'a pas été mis en proxy par le pare-feu.
- 2. Un Hello de client TLS provient du client avec l'ID IP 12083. Le paquet est mis en proxy par le pare-feu (le pare-feu, dans ce cas, a été configuré avec la stratégie de déchiffrement TLS) et l'ID IP est modifié à 52534. En outre, la somme de contrôle TCP du paquet est corrompue (en raison d'un défaut logiciel qui a été réparé par la suite).
- 3. Le pare-feu est en mode Proxy TCP et envoie un ACK au client (qui usurpe le serveur).

	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=2
	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
<						
5	Internet Protocol Version 4, Src:	192,168,202,11, D	st: 192.168.202.111	1		
~	Transmission Control Protocol, Sr	c Port: 15880, Dst	Port: 443, Seq: 24	486930708,	Ack: 3674485383, Len: 10	87
	Source Port: 15880					
	Destination Port: 443					
	[Stream index: 1]					
	[TCP Segment Len: 187]					
	Sequence number: 2486930708					
	[Next sequence number: 2486930	895]				
	Acknowledgment number: 3674405	383				
	1000 = Header Length: 32	bytes (8)				
	> Flags: 0x018 (PSH, ACK)					
	Window size value: 64					
	[Calculated window size: 8192]					
	[Window size scaling factor: 1]	28]				
	> Checksum: 0x0c65 incorrect, she	ould be 0x3063(mayb	be caused by "TCP o	checksum o	ffload"?)	
	[Checksum Status: Bad]					
	[Calculated Checksum: 0x3063]					
	Urgent pointer: 0					
	> Options: (12 bytes), No-Operat:	ion (NOP), No-Opera	stion (NOP), Timest	amps		
	> [SEQ/ACK analysis]					
	> [Timestamps]					
	TCP payload (187 bytes)					
1.3	Secure Sockets Laver					

- 4. Le pare-feu ne reçoit aucun paquet ACK TCP du serveur et retransmet le message Hello du client TLS. Cela est encore dû au mode Proxy TCP que le pare-feu a activé.
- 5. Après environ 30 secondes, le pare-feu abandonne et envoie un RST TCP au client.
- 6. Le pare-feu envoie un RST TCP au serveur.

Pour référence :

Traitement de la connexion TLS/SSL Firepower

Cas 10. Problème de connectivité HTTPS (scénario 2)

Description du problème : l'enregistrement de la licence Smart FMC échoue.

Overview Analysis Poli	cles Devices	Objects	AMP	Intelligence								Deploy	/ 🧛 Sys	stem Help 🔻	admin 🔻
						Configuration	Users	Domains	Integration	Updates	Licenses • Sr	mart Licenses	Health 🔻	Monitoring •	Tools
					Error Failed t the DN	o send the messa S Server/HTTP Pro	ge to the ser oxy settings.	ver. Please verify	×			Smart Licenses Registration Failed to regi	Dismis to the Cisco ster	s) Smart Softwar	× re Manag
Welcome to Smar Before you use Smart I from <u>Cisco Smart Soft</u> Smart License Status	t Licenses Licenses, obtain a vare Manager, th	a registration en click Regis	token ster	Regist	er										
Usage Authorization:															
Product Registration:	Unreg	istered													
Assigned Virtual Account:															
Export-Controlled Features:															
Cisco Success Network:															

Cette image présente la topologie :

192.168.0.100	

Flux affecté :

Adresse IP source : 192.168.0.100

Dst : tools.cisco.com

Protocole : TCP 443 (HTTPS)

Analyse de capture

Activez la capture sur l'interface de gestion FMC :

FMC	Capture on FMC eth0 (mgmt) interface	Cisco Licensing Portal
	192.168.0.100	

Réessayez de vous inscrire. Lorsque le message d'erreur apparaît, appuyez sur CTRL-C pour arrêter la capture :

```
<#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#</pre>
```

Collectez la capture à partir du FMC (System > Health > Monitor, sélectionnez le périphérique et sélectionnez Advanced Troubleshooting), comme illustré dans l'image :

Overview	Analysis	Policies	Devices	Objects	AMP	Intell	igence		Deploy	0 ₈ Sy	stem Help 🔻	admin 🔻
	Cor	nfiguration	Users	Domains	Integ	ration	Updates	Licenses 🔻	Health 🕨	Monitor	Monitoring •	Tools 🔻
Advance	ed Trou	blesho	oting									
File Downl	oad											
		File CA	Р.рсар									
					Dov	vnload	Back					

L'image montre la capture FMC sur Wireshark :

Eil	CAP.pcap jile Edit View Go Capture Analyze Statistics Telephony Wireless Iools Help										
L	Apply a display filter <ctrl-></ctrl->										
No	. Time	Source	Destination	Protocol	Length Info						
	1 2019-10-23 07:44:59.218797	192.168.0.100	10.229.20.96	TLSv1.2	2 107 Application Data						
	2 2019-10-23 07:44:59.220929	10.229.20.96	192.168.0.100	TLSv1.2	2 123 Application Data						
	3 2019-10-23 07:44:59.220960	192.168.0.100	10.229.20.96	TCP	54 443 → 64722 [ACK] Seq=1380971613 Ack=2615750168 Win=249 Len=0						
	4 2019-10-23 07:45:02.215376	192.168.0.100	10.229.20.96	TLSv1.2	2 107 Application Data						
	5 2019-10-23 07:45:02.217321	10.229.20.96	192.168.0.100	TLSv1.2	2 123 Application Data						
	6 2019-10-23 07:45:02.217336	192.168.0.100	10.229.20.96	TCP	54 443 → 64722 [ACK] Seq=1380971666 Ack=2615750237 Win=249 Len=0						
	7 2019-10-23 07:45:05.215460	192.168.0.100	10.229.20.96	TLSv1.2	2 107 Application Data						
	8 2019-10-23 07:45:05.217331	10.229.20.96	192.168.0.100	TLSv1.2	2 123 Application Data						
	9 2019-10-23 07:45:05.217345	192.168.0.100	10.229.20.96	TCP	54 443 → 64722 [ACK] Seq=1380971719 Ack=2615750306 Win=249 Len=0						
	10 2019-10-23 07:45:06.216584	10.229.20.96	192.168.0.100	TCP	66 64784 → 443 [SYN] Seq=4002690284 Win=64240 Len=0 MSS=1380 WS=256 S						
	11 2019-10-23 07:45:06.216631	192.168.0.100	10.229.20.96	TCP	66 443 → 64784 [SYN, ACK] Seq=3428959426 Ack=4002690285 Win=29200 Ler						
	12 2019-10-23 07:45:06.218550	10.229.20.96	192.168.0.100	TCP	60 64784 → 443 [ACK] Seq=4002690285 Ack=3428959427 Win=66048 Len=0						
	13 2019-10-23 07:45:06.219386	10.229.20.96	192.168.0.100	TLSv1.2	2 571 Client Hello						

Conseil : afin de vérifier toutes les nouvelles sessions TCP qui ont été capturées, utilisez le filtre d'affichage tcp.flags==0x2 sur Wireshark. Cela filtre tous les paquets TCP SYN qui ont été capturés.

	CAP.pcap									
Eile	ile Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help									
	▲ = Z @ = 1 1 2 4 + + = = + + = = = = = = = = = = = = =									
tcp	.flags==0x2									
No.	Time	Source	Destination	Protocol	Length Info					
Г	10 2019-10-23 07:45:06.216584	10.229.20.96	192.168.0.100	TCP	66 64784 → 443 [SYN] Seq=4002690284 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1					
1	19 2019-10-23 07:45:06.225743	10.229.20.96	192.168.0.100	TCP	66 64785 → 443 [SYN] Seq=3970528579 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1					
	45 2019-10-23 07:45:12.403280	10.229.20.96	192.168.0.100	TCP	66 64790 → 443 [SYN] Seq=442965162 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1					
	51 2019-10-23 07:45:12.409842	10.229.20.96	192.168.0.100	TCP	66 64791 → 443 [SYN] Seq=77539654 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1					
	72 2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74 35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16127801 TSecr=0 WS=128					
	108 2019-10-23 07:45:24.969622	192.168.0.100	72.163.4.38	TCP	74 35756 → 443 [SYN] Seq=1993860949 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16138303 TSecr=0 WS=128					
	137 2019-10-23 07:45:35.469403	192.168.0.100	173.37.145.8	TCP	74 58326 → 443 [SYN] Seq=723413997 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2040670996 TSecr=0 WS=128					
	163 2019-10-23 07:45:45.969384	192.168.0.100	173.37.145.8	TCP	74 58330 → 443 [SYN] Seq=2299582550 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2040681496 TSecr=0 WS=128					
	192 2019-10-23 07:45:56.468604	192.168.0.100	72.163.4.38	TCP	74 35768 → 443 [SYN] Seq=1199682453 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16169802 TSecr=0 WS=128					
	227 2019-10-23 07:46:07.218984	10.229.20.96	192.168.0.100	TCP	66 64811 → 443 [SYN] Seq=1496581075 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1					
	236 2019-10-23 07:46:07.225881	10.229.20.96	192.168.0.100	TCP	66 64812 → 443 [SYN] Seq=563292608 Win=64240 Len=0 MSS=1380 WS=256 SACK_PERM=1					

 \wp Conseil : appliquez en tant que colonne le champ Server Name du paquet SSL Client Hello.

75 2019-10-23 07:45:14.634091 192.168.0.1	00 72.163.4.38	TLSv1.2	571 Client Hello	
<				
> Frame 75: 571 bytes on wire (4568 bits), 571 b > Ethernet II, Src: Vmware_10:d0:a7 (00:0c:29:10 > Internet Protocol Version 4, Src: 192.168.0.10 > Transmission Control Protocol, Src Port: 35752 Y Secure Sockets Laver	ytes captured (4568 bits) :d0:a7), Dst: Cisco_f6:1d 0, Dst: 72.163.4.38 , Dst Port: 443, Seq: 242	:ae (00:b	e:75:f6:1d:ae) Ack: 2770078885, Len: 517	
 TLSv1.2 Record Layer: Handshake Protocol Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 	Expand Subtrees Collapse Subtrees Expand All Collapse All			
 Handshake Protocol: Client Hello Handshake Tune: Client Hello (1) 	Apply as Column			
<pre>kanoshake type: Crient Heiio (1) Length: 508 Version: TLS 1.2 (0x0303) Random: 234490a107438c73b595646532; Session ID Length: 0 Cipher Suites Length: 100 Cipher Suites (50 wites)</pre>	Apply as Filter Prepare a Filter Conversation Filter Colorize with Filter Follow	• • •		
Compression Methods Length: 1 > Compression Methods (1 method) Extensions Length: 367	Copy Show Packet Bytes Export Packet Bytes	•		
 Extension: server_name (len=20) Type: server_name (0) Length: 20 Server Name Indication extension 	Wiki Protocol Page Filter Field Reference Protocol Preferences			
Server Name list length: 18 Server Name Type: host_name (Decode As Go to Linked Packet			
Server Name: tools.cisco.com	Show Linked Packet in New Wind	ow		

Conseil : appliquez ce filtre d'affichage pour afficher uniquement les messages Hello du client ssl.handshake.type == 1

📕 ssl.har	andshake.type == 1						
No.	Time	Source	Destination	Protocol	Length	Server Name	Info
1	13 2019-10-23 07:45:06.219386	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
2	23 2019-10-23 07:45:06.227250	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
4	48 2019-10-23 07:45:12.406366	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
5	54 2019-10-23 07:45:12.412199	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
7	75 2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
11	11 2019-10-23 07:45:25.136089	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
14	40 2019-10-23 07:45:35.637252	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
16	66 2019-10-23 07:45:46.136858	192.168.0.100	173.37.145.8	TLSv1.2	571	tools.cisco.com	Client Hello
19	95 2019-10-23 07:45:56.635438	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello
23	30 2019-10-23 07:46:07.221567	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello
24	40 2019-10-23 07:46:07.228486	10.229.20.96	192.168.0.100	TLSv1.2	571		Client Hello

Remarque : au moment de la rédaction de ce document, le portail Smart Licensing (tools.cisco.com) utilise les adresses IP suivantes : 72.163.4.38, 173.37.145.8

Suivez l'un des flux TCP (Follow > TCP Stream), comme illustré dans l'image.

75 2019-10-23 07:45:14.634091 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.cc 111 2019-10-23 07:45:25.136089 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.cc 140 2019-10-23 07:45:25.136089 192.168.0.100 773.37.145.8 TLSv1.2 571 tools.cisco.cc 155 2019-10-23 07:45:46.13685 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.cc 195 2019-10-23 07:45:46.13685 192.168.0.100 72.163.4.38 TLSv1.2 571 tools.cisco.cc 195 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 bthernet II, Src: Wmware_10:d0:a7 (00:6:29:10:d0:a7), Dst: Cisco_f6:1d:ae (00:be:75:f6:1d:ae) SCP SCP nternet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38 2027943532, Ack: 2770078885, Let Copy							
Conversation Filter Apply as Filter Prepare a Filter Prepare a Filter Conversation Filter Conversation Filter Conversation Control Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38 ScTP ransmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Lee Follow TCP Stream Y TLSV1.2 Record Layer: Handshake Protocol: Client Hello Copy UDP Stream Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Show Packet in New Window	75 2019-10-23 07:45:14.634091 111 2019-10-23 07:45:25.136089 140 2019-10-23 07:45:35.637252 166 2019-10-23 07:45:46.136858 195 2019-10-23 07:45:56.635438 230 2019-10-23 07:46:07.221567 240 2019-10-23 07:46:07.228486	192.168.0.100 192.168.0.100 192.168.0.100 192.168.0.100 192.168.0.100 192.168.0.100 10.229.20.96 10.229.20.96	72.163.4.38 72.163.4.38 173.37.145.8 173.37.145.8 72.163.4.38 192.168.0.100 192.168.0.100	TLSv1.2 TLSv1.2 TLSv1.2 TLSv1.2 TLSv1.2 TLSv1.2 TLSv1.2 TLSv1.2	571 tools.cisco.cc 571 tools.cisco.cc 571 tools.cisco.cc 571 tools.cisco.cc 571 tools.cisco.cc 571 tools.cisco.cc 571	Mark/Unmark Packet Ignore/Unignore Packet Set/Unset Time Reference Time Shift Packet Comment	
nternet Protocol Version 4, Src: 192.168.0.100, Dst: 72.163.4.38 ransmission Control Protocol, Src Port: 35752, Dst Port: 443, Seq: 2427943532, Ack: 2770078885, Le ceure Sockets Layer TLSv1.2 Record Layer: Handshake Protocol: Client Hello Content Type: Handshake (22) Version: TLS 1.0 (0x0301) TCP Stream HTTP Stream HTTP Stream Show Packet in New Window	rame 75: 571 bytes on wire (4568 thernet II, Src: Vmware_10:d0:a7	Apply as Filter Prepare a Filter Conversation Filter Colorize Conversation SCTP					
	<pre>nternet Protocol Version 4, Src: ransmission Control Protocol, Src ecure Sockets Layer < TLSv1.2 Record Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301)</pre>	192.168.0.100, Ds Port: 35752, Dst Protocol: Client	t: 72.163.4.38 : Port: 443, Seq: 2 Hello	427943532, /	lck: 2770078885, Le	Follow Copy Protocol Preferences Decode As Show Packet in New Window	TCP Stream UDP Stream SSL Stream HTTP Stream

E - · Expre

No.	Time	Source	Destination	Protocol	Length	Server Name	Infe			
- 7	72 2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74		35752 → 443 [SYN] Seg=2427943531 Win=29200 Len=0 MSS=1460 SACK PERM=1 TSval=16127801 TSecr=0 WS=128			
7	73 2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60	(()	443 → 35752 [SYN, ACK] Seg=2770078884 Ack=2427943532 Win=8190 Len=0 MSS=1330			
7	74 2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seg=2427943532 Ack=2770078885 Win=29200 Len=0			
7	75 2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571	tools.cisco.com	Client Hello			
7	76 2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=32768 Len=0			
7	77 2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150		Server Hello			
7	78 2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=29200 Len=0			
7	79 2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1384		443 → 35752 [PSH, ACK] Seq=2770078981 Ack=2427944049 Win=32768 Len=1330 [TCP segment of a reassembled PDU]			
8	80 2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=31920 Len=0			
8	81 2019-10-23 07:45:14.96682	72.163.4.38	192.168.0.100	TLSv1.2	155	4	Certificate			
8	82 2019-10-23 07:45:14.9668	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080412 Win=31920 Len=0			
8	33 2019-10-23 07:45:14.966915	72.163.4.38	192.168.0.100	TLSv1.2	63		Server Hello Done			
8	34 2019-10-23 07:45:14.966925	192.168.0.100	72.163.4.38	TCP	54		35752 → 443 [ACK] Seq=2427944049 Ack=2770080421 Lin=31920 Len=0			
8	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) 5			
8	86 2019-10-23 07:45:14.967261	192.168.0.100	72.163.4.38	TCP	54	6				
8	37 2019-10-23 07:45:14.967382	72.163.4.38	192.168.0.100	TCP	60		443 → 35752 [ACK] Seq=2770080421 Ack=2427944056 Win=32768 Len=0			
Lε	88 2019-10-23 07:45:14.967398	192.168.0.100	72.163.4.38	тср	54		35752 → 443 [RST] Seq=2427944056 Win=0 Len=0			
<										
> Fra	Frame 75: 571 hutes on wire (AS68 hits) 571 hutes cantured (AS68 hits)									
) Frame /3: 3/1 dytes on wire (4008 Dits), 3/1 dytes captured (4008 Dits) Schwarzet TJ. Serv Humanen 10:40:47 (400:40:10) Her (Schwarz) (5:65 fild:an)									
> Eth	ernet II, Src: Vmware_10:d0:a7	(00:0c:29:10:d0:a7), Dst: Cisco_f6:1	d:ae (00:	be:75:f	6:1d:ae)				
> Eth > Int	ernet II, Src: Vmware_10:d0:a7 ernet Protocol Version 4, Src: 1	(00:0c:29:10:d0:a7 192.168.0.100, Dst), Dst: Cisco_f6:1 : 72.163.4.38	d:ae (00:	be:75:f	6:1d:ae)				
> Eth > Int > Tra	ernet II, Src: Vmware_10:d0:a7 ernet Protocol Version 4, Src: nsmission Control Protocol, Src	(00:0c:29:10:d0:a7 192.168.0.100, Dst Port: 35752, Dst), Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24	id:ae (00:	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra ¥ Sec	ernet II, Src: Vmware_10:d0:a7 ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer	(00:0c:29:10:d0:a7 192.168.0.100, Dst Port: 35752, Dst), Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24	d:ae (00:	be:75:f Ack: 2	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	ernet II, Src: Vmare_10:d0:a7 ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer LSVI.2 Record Layer: Handshake	(00:0c:29:10:d0:a7 192.168.0.100, Dst Port: 35752, Dst Protocol: Client	(1), Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24 Hello	d:ae (00:1	be:75:f Ack: 2	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra * Sec	ernet II, Src: Vmare_l0:d0:a7 ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer TLSv1.2 Record Layer: Handshake Content Type: Handshake (22)	(00:0c:29:10:d0:a7 192.168.0.100, Dst Port: 35752, Dst Protocol: Client), Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24 Hello	ld:ae (00:1	be:75:f Ack: 2	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	ernet II, Srit Gyles on Wile (4300 ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer TLSV1.2 Record Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x8301)	(00:0c:29:10:00:a7 192.168.0.100, Dst Port: 35752, Dst Protocol: Client), Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24 Hello	ld:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	ernet II, Sric Vieware 21:00:30 ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer TLSV1.2 Record Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512	(00:0c:29:10:d0:a7 192.168.0.100, Dst Port: 35752, Dst Protocol: Client), Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24 Hello	d:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	ernet II, Sric Vieware 21:60:37 ernet Protocol Version 4, Src: smäission Control Protocol, Src ure Sockets Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 V Handshake Protocol: Client He	(00:0c:29:10:00:07 192.168.0.100, Dst Port: 35752, Dst Protocol: Client), Dst: Cisco_f6:1 :: 72.163.4.38 Port: 443, Seq: 24 Hello	d:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	<pre>merrer II, Sric Vmare [10:60:a7 ernet Protocol Version 4, Src:: nsmission Control Protocol, Src ure Sockets Layer IISv1.2 Record Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 5120col: Client He Handshake Protocol: Client He</pre>	(00),02:03:04:07 192:168:0.100, Dst Port: 35752, Dst Protocol: Client *110 10 (1)), Dst: Cisco_f6:] : 72.163.4.38 Port: 443, Seq: 24 Hello	d:ae (00:1	Ack: 2	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	<pre>werp. JJ. Syles on wife (also) ernet IP, Src: Vmware_10:40:a7 ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer TLSV1.2 Record Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 >> Handshake Protocol: Client He Handshake Type: Client Hel Length: 508</pre>	(06)(05)(05)(06)(06)(06)(06)(06)(06)(06)(06)(06)(06	J, Dst: Cisco_F6:: : 72.163.4.38 Port: 443, Seq: 2ℓ Hello	d:ae (00:1	Ack: 2	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra • Sec	ernet II, Sric Vieware 1(:3:00); ernet Protocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 V Handshake Protocol: Client Hel Handshake Type: Client Hel Length: 508 Version: TLS 1.2 (0x0303)	(06)(02)(05)(06)(07) 192.168.0.100, Dst Port: 35752, Dst Protocol: Client	J, Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 2ℓ Hello	d:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	 mar J. J. Dyles on Mile (2:00:a7 ernet IProtocol Version 4, Src: nsmission Control Protocol, Src ure Sockets Layer TLSV1.2 Record Layer: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 Handshake Protocol: Client He Handshake Protocol: Client He Length: 508 Version: TLS 1.2 (0x0303) Random: 2344090107438c7305 	00165), 571 05/05 0 (001-062-2011-05-05 192_168_0.100, Dst Port: 35752, Dst Protocol: Client ello 10 (1)	<pre>b, Dst: Cisco_f6:1 : 72.163.4.38 Port: 443, Seq: 24 Hello bb7ac16897184</pre>	d:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra > Sec	ernet II, Sric Vimare (1:30:a7) ernet Protocol Version 4, Src: smässion Control Protocol, Src ure Sockets Layer: Handshake Content Type: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 V Handshake Protocol: Client He Handshake Type: Client He Length: 508 Version: TLS 1.2 (0x0303) > Random: 234490a107438c7305 Session ID Length: 0	(06) 06:29:10:00:a7 192.168.0.100, Dst Port: 35752, Dst Protocol: Client ello 10 (1) 3	<pre>>, Dst: Cisco_f6:: : 72.163.4.38 Port: 443, Seq: 2/ Hello bb7ac16897184</pre>	d:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			
> Eth > Int > Tra • Sec	 mar J. J. Gyles on Wile (2004) ernet IT, Sric Yumare [10:00:a7 ernet Protocol Version 4, Src:: mainsion Control Protocol, Src ure Sockets Layer TLS-1.2 Record Layer: Handshake (22) Version: TLS 1.0 (0x0301) Length: 512 Handshake Protocol: Client Hel Length: 508 Version: TLS 1.2 (0x0303) Random: 234490a107438c73b5 Session ID Length: 0 Cipher Suites Length: 100 	(06) (02) (06) (06) (06) (06) (06) (06) (06) (06	<pre>), Dst: Cisco_f6:: : 72.163.4.38 Port: 443, Seq: 24 Hello bb7ac16897184</pre>	d:ae (00:1	be:75:f	6:1d:ae) 770078885, Len: 5	7			

Principaux points :

tro stream on 5

- 1. Il existe une connexion TCP en trois étapes.
- 2. Le client (FMC) envoie un message Hello de client SSL vers le portail Smart Licensing.
- 3. L'ID de session SSL est 0. Cela signifie qu'il ne s'agit pas d'une reprise de session.
- 4. Le serveur de destination répond avec le message Server Hello, Certificate et Server Hello Done.
- 5. Le client envoie une alerte SSL fatale qui concerne une « CA inconnue ».
- 6. Le client envoie un RST TCP pour fermer la session.
- 7. La durée totale de la session TCP (de l'établissement à la fermeture) était d'environ 0,5 seconde.

Sélectionnez le certificat de serveur et développez le champ de l'émetteur pour voir le commonName. Dans ce cas, le nom commun indique un périphérique qui fait de l'homme du milieu (MITM).

No.	Time	Source	Destination	Protocol	Length Server Name	Info
-	72 2019-10-23 07:45:14.466836	192.168.0.100	72.163.4.38	TCP	74	35752 → 443 [SYN] Seq=2427943531 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=16127801
	73 2019-10-23 07:45:14.632885	72.163.4.38	192.168.0.100	TCP	60	443 → 35752 [SYN, ACK] Seq=2770078884 Ack=2427943532 Win=8190 Len=0 MSS=1330
	74 2019-10-23 07:45:14.632935	192.168.0.100	72.163.4.38	TCP	54	35752 → 443 [ACK] Seq=2427943532 Ack=2770078885 Win=29200 Len=0
	75 2019-10-23 07:45:14.634091	192.168.0.100	72.163.4.38	TLSv1.2	571 tools.cisco.com	Client Hello
	76 2019-10-23 07:45:14.634796	72.163.4.38	192.168.0.100	TCP	60	443 → 35752 [ACK] Seq=2770078885 Ack=2427944049 Win=32768 Len=0
	77 2019-10-23 07:45:14.966729	72.163.4.38	192.168.0.100	TLSv1.2	150	Server Hello
	78 2019-10-23 07:45:14.966772	192.168.0.100	72.163.4.38	TCP	54	35752 → 443 [ACK] Seq=2427944049 Ack=2770078981 Win=29200 Len=0
÷	79 2019-10-23 07:45:14.966834	72.163.4.38	192.168.0.100	TCP	1384	443 → 35752 [PSH, ACK] Seq=2770078981 Ack=2427944049 Win=32768 Len=1330 [TCP segment
	80 2019-10-23 07:45:14.966850	192.168.0.100	72.163.4.38	TCP	54	35752 → 443 [ACK] Seq=2427944049 Ack=2770080311 Win=31920 Len=0
+	81 2019-10-23 07:45:14.966872	72.163.4.38	192.168.0.100	TLSv1.2	155	Certificate
<						
	Length: 1426					
	✓ Handshake Protocol: Certific	ate				
	Handshake Type: Certificat	te (11)				
	Length: 1422					
	Certificates Length: 1419					
	 Certificates (1419 bytes) 					
	Certificate Length: 141	16				
	 Certificate: 3082058436 	082046ca0030201020	020d00aa23af5d607e0	000 (id	d-at-commonName=tools.cise	o.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose,id-at-sta:
	✓ signedCertificate					
	version: v3 (2)					
	serialNumber: 0x0	0aa23af5d607e0000	2f423880			
	> signature (sha256	WithRSAEncryption)			
	✓ issuer: rdnSequen	ice (0)				
	✓ rdnSequence: 3	items (id-at-com	monName=FTD4100_MIT	M,id-at-o	rganizationalUnitName=FTD	_OU,id-at-organizationName=FTD_O)
	> RDNSequence	item: 1 item (id-	-at-organizationNam	e=FTD_0)		
	> RDNSequence	item: 1 item (id-	-at-organizationalU	nitName=F1	TD_OU)	
	> RDNSequence	item: 1 item (id-	-at commonName=FTD4	100_MITM)		
	> validity				-	
	> subject: rdnSeque	nce (0)				
	> subjectPublicKeyI	info				
	✓ extensions: 6 ite	ms				

Ceci est montré dans cette image :



Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Effectuez des captures supplémentaires.

Effectuez des captures sur le périphérique de pare-feu de transit :



CAPI montre :

. t	p.stre	am eq 57													
No.		Time			Source		Destination		Protocol	Length	Server Name	Info			
-	1221	2019-10-2	2 17:49	:03.212681	192.16	8.0.100	173.37.	145.8	TCP	74		39924 → 443	[SYN]	Seq=427175838 Win=29200 Len=0 MSS=146	0 SACK_PERM=1
	1222	2019-10-2	2 17:49	03.379023	173.37	7.145.8	192.168	.0.100	TCP	58		443 → 39924	[SYN,	ACK] Seq=236460465 Ack=427175839 Win=	8190 Len=0 MSS
	1223	2019-10-2	2 17:49	03.379298	192.16	8.0.100	173.37.	145.8	TCP	54		39924 → 443	[ACK]	Seq=427175839 Ack=236460466 Win=29200	Len=0
	1224	2019-10-2	2 17:49	0:03.380336	192.16	8.0.100	173.37.	145.8	TLSv1.2	571	tools.cisco.com	Client Hello)		
	1225	2019-10-2	2 17:49	03.380732	173.37	145.8	192.168	.0.100	TCP	54		443 → 39924	[ACK]	Seq=236460466 Ack=427176356 Win=32768	Len=0
	1226	2019-10-2	2 17:49	0:03.710092	173.37	145.8	192.168	.0.100	TLSv1.2	150		Server Hello)		
+	1227	2019-10-2	2 17:49	0:03.710092	173.37	145.8	192.168	.0.100	TCP	1384		443 → 39924	[PSH,	ACK] Seq=236460562 Ack=427176356 Win=	32768 Len=1330
+	1228	2019-10-2	2 17:49	9:03.710092	173.37	145.8	192.168	.0.100	TLSv1.2	155		Certificate			
	1229	2019-10-2	2 17:49	9:03.710107	173.37	145.8	192.168	.0.100	TLSv1.2	63		Server Hello	Done		
	1230	2019-10-2	2 17:49	03.710412	192.16	8.0.100	173.37.	145.8	TCP	54		39924 → 443	[ACK]	Seq=427176356 Ack=236460562 Win=29200	Len=0
	1231	2019-10-2	2 17:49	9:03.710519	192.16	8.0.100	173.37.	145.8	TCP	54		39924 → 443	[ACK]	Seq=427176356 Ack=236461892 Win=31920	Len=0
	1232	2019-10-2	2 17:49	0:03.710519	192.16	8.0.100	173.37.	145.8	TCP	54		39924 → 443	[ACK]	Seq=427176356 Ack=236461993 Win=31920	Len=0
	1233	2019-10-2	2 17:49	0:03.710534	192.16	8.0.100	173.37.	145.8	TCP	54		39924 → 443	[ACK]	Seq=427176356 Ack=236462002 Win=31920	Len=0
	1234	2019-10-2	2 17:49	9:03.710626	192.16	8.0.100	173.37.	145.8	TLSv1.2	61		Alert (Level	l: Fat	al, Description: Unknown CA)	
	1235	2019-10-2	2 17:49	9:03.710641	173.37	145.8	192.168	.0.100	TCP	54		443 → 39924	[ACK]	Seq=236462002 Ack=427176363 Win=32768	Len=0
	1236	2019-10-2	2 17:49	9:03.710748	192.16	8.0.100	173.37.	145.8	тср	54		39924 → 443	[RST,	ACK] Seq=427176363 Ack=236462002 Win=	31920 Len=0
L.	1237	2019-10-2	2 17:49	9:03.710870	192.16	8.0.100	173.37.	145.8	TCP	54		39924 → 443	[RST]	Seq=427176363 Win=0 Len=0	
<															
		Length: 3	1426												
	~	Handshak	Proto	col: Certific	ate										
		Handsh	ake Typ	pe: Certifica	te (11)										
		Length	: 1422												
		Certif	icates	Length: 1419											
		✓ Certif	icates	(1419 bytes)											
		Cer	tificat	te Length: 14	16										
		✓ Cer	tificat	te: 308205843	082046ca	0030201020	20d00aa23	af5d607e000	0 (id	l-at-co	mmonName=tools.ci	sco.com,id-at-or	ganiz	ationName=Cisco Systems, Inc.,id-at-lo	calityName=San
		~	signedC	ertificate											
			vers	ion: v3 (2)											
			seri	alNumber: 0x0	0aa23af	5d607e0000	2f423880								
			> sign	ature (sha256	WithRSA	Encryption)								
			✓ issu	er: rdnSequer	ce (0)										
			V re	dnSequence: 3	items	(id-at-com	monName=F1	D4100_MITM	id-at-o	rganiza	tionalUnitName=FT	D_OU,id-at-orga	nizati	lonName=FTD_0)	
			>	RDNSequence	item:	1 item (id-	-at-organi	zationName=	FTD_0)						
			>	RDNSequence	item: :	1 item (id-	-at-organi	zationalUni	tName=F1	(U0_D1					
			>	RDNSequence	item: :	1 item (id-	-at-common	Name=FTD410	(MTIM_0						
			> vali	dity											

CAPO montre :

	tcp.stream eq 57					
N	. Time	Source	Destination	Protocol	Length Server Name	Info
5	1169 2019-10-22 17:49:03.212849	192.168.0.100	173.37.145.8	TCP	78	39924 → 443 [SYN] Seq=623942018 Win=29200 Len=0 MSS=1380 SACK_PERM=1 TSval
	1170 2019-10-22 17:49:03.378962	173.37.145.8	192.168.0.100	TCP	62	443 → 39924 [SYN, ACK] Seq=4179450724 Ack=623942019 Win=8190 Len=0 MSS=1330
Π	1171 2019-10-22 17:49:03.379329	192.168.0.100	173.37.145.8	TCP	58	39924 → 443 [ACK] Seq=623942019 Ack=4179450725 Win=29200 Len=0
	1172 2019-10-22 17:49:03.380793	192.168.0.100	173.37.145.8	TLSv1.2	512 tools.cisco.com	m Client Hello
+	1173 2019-10-22 17:49:03.545748	173.37.145.8	192.168.0.100	TCP	1388	443 → 39924 [PSH, ACK] Seq=4179450725 Ack=623942473 Win=34780 Len=1330 [TC
+	1174 2019-10-22 17:49:03.545809	173.37.145.8	192.168.0.100	TCP	1388	443 → 39924 [PSH, ACK] Seq=4179452055 Ack=623942473 Win=34780 Len=1330 [TC
	1175 2019-10-22 17:49:03.545824	192.168.0.100	173.37.145.8	TCP	58	39924 → 443 [ACK] Seq=623942473 Ack=4179453385 Win=65535 Len=0
÷	1176 2019-10-22 17:49:03.545915	173.37.145.8	192.168.0.100	TCP	1388	443 → 39924 [PSH, ACK] Seq=4179453385 Ack=623942473 Win=34780 Len=1330 [TC
÷	1177 2019-10-22 17:49:03.545961	173.37.145.8	192.168.0.100	TCP	1388	443 → 39924 [PSH, ACK] Seq=4179454715 Ack=623942473 Win=34780 Len=1330 [TC
	1178 2019-10-22 17:49:03.545961	192.168.0.100	173.37.145.8	TCP	58	39924 → 443 [ACK] Seq=623942473 Ack=4179456045 Win=65535 Len=0
+	1179 2019-10-22 17:49:03.709420	173.37.145.8	192.168.0.100	TLSv1.2	82	Server Hello, Certificate, Server Hello Done
	1180 2019-10-22 17:49:03.710687	192.168.0.100	173.37.145.8	TLSv1.2	65	Alert (Level: Fatal, Description: Unknown CA)
	1181 2019-10-22 17:49:03.710885	192.168.0.100	173.37.145.8	TCP	58	39924 → 443 [FIN, PSH, ACK] Seq=623942480 Ack=4179456069 Win=65535 Len=0
Ľ	1182 2019-10-22 17:49:03.874542	173.37.145.8	192.168.0.100	TCP	58	443 → 39924 [RST, ACK] Seq=4179456069 Ack=623942480 Win=9952 Len=0
<						
	Length: 5339					
	> Handshake Protocol: Server H	Hello				
	✓ Handshake Protocol: Certifie	cate				
	Handshake Type: Certifica	ate (11)				
	Length: 5240					
	Certificates Length: 5237	7				
	 Certificates (5237 bytes))				
	Certificate Length: 20	25				
	 Certificate: 308207e53 	08205cda003020102	2143000683b0f7504f	7b2 (id	-at-commonName=tools.	cisco.com,id-at-organizationName=Cisco Systems, Inc.,id-at-localityName=San Jose
	> signedCertificate					
	> algorithmIdentifier	(sha256WithRSAEnd	ryption)			
	Padding: 0					
	encrypted: 6921d084	f7a6f6167058f14e2a	ad8b98b4e6c971ea6e	a3b4		
	Certificate Length: 17	36				
	 Certificate: 308206c43 	08204aca003020102	2147517167783d0437	'eb5 (id	l-at-commonName=Hydran	tID SSL ICA G2, id-at-organizationName=HydrantID (Avalanche Cloud Corporation), id
	✓ signedCertificate					
	version: v3 (2)					
	serialNumber: 0x	7517167783d0437eb5	56c357946e4563b8eb	d3ac		
	> signature (sha25)	6WithRSAEncryption)			
	✓ issuer: rdnSeque	nce (0)				
	> rdnSequence:	3 items (id-at-com	monName=QuoVadis R	oot CA 2,io	d-at-organizationName=	-QuoVadis Limited,id-at-countryName=BM)
	> validity					

Ces captures prouvent que le pare-feu de transit modifie le certificat de serveur (MITM)

Action 2. Vérifiez les journaux des périphériques.

Vous pouvez collecter l'offre groupée FMC TS comme décrit dans ce document :

https://www.cisco.com/c/en/us/support/docs/security/sourcefire-defense-center/117663-technote-SourceFire-00.html Dans ce cas, le fichier /dir-archives/var-log/process_stdout.log affiche des messages comme ceci :

<#root>

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

Solution recommandée

Désactivez le MITM pour le flux spécifique afin que FMC puisse s'enregistrer avec succès sur le cloud de licences Smart.

Cas 11 . Problème de connectivité IPv6

Description du problème : les hôtes internes (situés derrière l'interface INSIDE du pare-feu) ne peuvent pas communiquer avec les hôtes externes (hôtes situés derrière l'interface OUTSIDE du pare-feu).

Cette image présente la topologie :



Flux affecté :

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

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

Protocole : tout

Analyse de capture

Activer les captures sur le moteur FTD LINA.

<#root>

firepower#

capture CAPI int INSIDE match ip any6 any6



capture CAPO int OUTSIDE match ip any6 any6



Captures - Scénario non fonctionnel

Ces captures ont été effectuées en parallèle avec un test de connectivité ICMP de l'IP fc00:1:1:1::100 (routeur interne) à l'IP fc00:1:1:2::2 (routeur en amont).

La capture sur l'interface INSIDE du pare-feu contient :

No.	Time	Source	Destination	Protocol Angth Info
	1 2019-10-24 13:02:07.001663	fc00:1:1:1:100	ff02::1:ff00:1	ICMPv6 86 Neighbor Solicitation for fc00:1:1:1:1 from 4c:4e:35:fc:fc:d8
	2 2019-10-24 13:02:07.001876	fc00:1:1:1::1	fc00:1:1:1::100	ICMPv6 2 86 Neighbor Advertisement fc00:1:1:1::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:ae
	3 2019-10-24 13:02:07.002273	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 🚬 114 Echo (ping) request id=0x160d, seq=0, hop limit=64 (no response found!)
	4 2019-10-24 13:02:08.997918	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 3114 Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
	5 2019-10-24 13:02:10.998056	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 114 Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
	6 2019-10-24 13:02:11.999917	fe80::2be:75ff:fef6:1dae	fc00:1:1:1::100	ICMPv6 4 86 Neighbor Solicitation for fc00:1:1:1:::100 from 00:be:75:f6:1d:ae
	7 2019-10-24 13:02:12.002075	fc00:1:1:1::100	fe80::2be:75ff:fef6:1dae	ICMPv6 5 78 Neighbor Advertisement fc00:1:1:1:::100 (rtr, sol)
	8 2019-10-24 13:02:12.998346	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 114 Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
	9 2019-10-24 13:02:14.998483	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 6114 Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
	10 2019-10-24 13:02:17.062725	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1dae	ICMPv6 🥗 86 Neighbor Solicitation for fe80::2be:75ff:fef6:1dae from 4c:4e:35:fc:fc:d8
	11 2019-10-24 13:02:17.062862	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fefc:fcd8	ICMPv6 78 Neighbor Advertisement fe80::2be:75ff:fef6:1dae (rtr, sol)
	12 2019-10-24 13:02:22.059994	fe80::2be:75ff:fef6:1dae	fe80::4e4e:35ff:fefc:fcd8	ICMPv6 86 Neighbor Solicitation for fe80::4e4e:35ff:fefc:fcd8 from 00:be:75:f6:1d:ae
	13 2019-10-24 13:02:22.063000	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1dae	ICMPv6 78 Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

Principaux points :

- 1. Le routeur envoie un message de sollicitation de voisin IPv6 et demande l'adresse MAC du périphérique en amont (IP fc00:1:1::1).
- 2. Le pare-feu répond avec une annonce de voisin IPv6.
- 3. Le routeur envoie une requête d'écho ICMP.
- 4. Le pare-feu envoie un message de sollicitation de voisin IPv6 et demande l'adresse MAC du périphérique en aval (fc00:1:1:1::100).
- 5. Le routeur répond avec une annonce de voisin IPv6.
- 6. Le routeur envoie des requêtes d'écho ICMP IPv6 supplémentaires.

La capture sur l'interface EXTERNE du pare-feu contient :

No.	Time	Source	Destination	Protocol ath Info
1	1 2019-10-24 13:02:07.002517	fe80::2be:75ff:fef6:1d8e	ff02::1:ff00:2	ICMp 90 Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
	2 2019-10-24 13:02:07.005569	fc00:1:1:2::2	fe80::2be:75ff:fef6:1d8e	ICM 2 90 Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
	3 2019-10-24 13:02:08.997995	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 3 18 Echo (ping) request id=0x160d, seq=1, hop limit=64 (no response found!)
	4 2019-10-24 13:02:09.001815	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6 90 Neighbor Solicitation for fc00:1:1:1:1:100 from 4c:4e:35:fc:fc:d8
	5 2019-10-24 13:02:10.025938	fc00:1:1:2::2	ff02::1:ff00:100	ICMPys 4 90 Neighbor Solicitation for fc00:1:1:1:1:100 from 4c:4e:35:fc:fc:d8
	6 2019-10-24 13:02:10.998132	fc00:1:1:1::100	fc00:1:1:2::2	ICM 5 118 Echo (ping) request id=0x160d, seq=2, hop limit=64 (no response found!)
1.1.1	7 2019-10-24 13:02:11.050015	fc00:1:1:2::2	ff02::1:ff00:100	ICMPV6 6 90 Neighbor Solicitation for fc00:1:1:1:1:100 from 4c:4e:35:fc:fc:d8
3	8 2019-10-24 13:02:12.066082	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6 90 Neighbor Solicitation for fe80::2be:75ff:fef6:1d8e from 4c:4e:35:fc:fc:d8
	9 2019-10-24 13:02:12.066234	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6 82 Neighbor Advertisement fe80::2be:75ff:fef6:1d8e (rtr, sol)
1	0 2019-10-24 13:02:12.998422	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 118 Echo (ping) request id=0x160d, seq=3, hop limit=64 (no response found!)
1	1 2019-10-24 13:02:13.002105	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6 90 Neighbor Solicitation for fc00:1:1:1:1:100 from 4c:4e:35:fc:fc:d8
1	2 2019-10-24 13:02:14.090251	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6 90 Neighbor Solicitation for fc00:1:1:1::100 from 4c:4e:35:fc:fc:d8
1	3 2019-10-24 13:02:14.998544	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6 118 Echo (ping) request id=0x160d, seq=4, hop limit=64 (no response found!)
1	4 2019-10-24 13:02:15.178350	fc00:1:1:2::2	ff02::1:ff00:100	ICMPv6 90 Neighbor Solicitation for fc00:1:1:1:1:100 from 4c:4e:35:fc:fc:d8
1	5 2019-10-24 13:02:17.059963	fe80::2be:75ff:fef6:1d8e	fe80::4e4e:35ff:fefc:fcd8	ICMPv6 90 Neighbor Solicitation for fe80::4e4e:35ff:fefc:fc48 from 00:be:75:f6:1d:8e
1	6 2019-10-24 13:02:17.062512	fe80::4e4e:35ff:fefc:fcd8	fe80::2be:75ff:fef6:1d8e	ICMPv6 82 Neighbor Advertisement fe80::4e4e:35ff:fefc:fcd8 (rtr, sol)

Principaux points :

- 1. Le pare-feu envoie un message de sollicitation de voisin IPv6 qui demande l'adresse MAC du périphérique en amont (IP fc00:1:1:2::2).
- 2. Le routeur répond avec une annonce de voisin IPv6.
- 3. Le pare-feu envoie une requête d'écho ICMP IPv6.
- 4. Le périphérique en amont (routeur fc00:1:1:2::2) envoie un message de sollicitation de voisin IPv6 qui demande l'adresse MAC de l'adresse IPv6 fc00:1:1:1:1:100.
- 5. Le pare-feu envoie une requête d'écho ICMP IPv6 supplémentaire.
- 6. Le routeur en amont envoie un message de sollicitation de voisin IPv6 supplémentaire qui demande l'adresse MAC de l'adresse IPv6 fc00:1:1:1::100.

Le point 4 est très intéressant. Normalement, le routeur en amont demande l'adresse MAC de l'interface OUTSIDE du pare-feu (fc00:1:1:2::2), mais à la place, il demande l'adresse fc00:1:1:1::100. Ceci indique une erreur de configuration.

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Vérifiez la table de voisinage IPv6.

La table de voisinage IPv6 du pare-feu est correctement remplie.

<#root>

firepower#

show ipv6 neighbor | i fc00

fc00:1:1:2::2 fc00:1:1:1::100 58 4c4e.35fc.fcd8 STALE OUTSIDE 58 4c4e.35fc.fcd8 STALE INSIDE

Action 2. Vérifiez la configuration IPv6.

Voici la configuration du pare-feu.

<#root>
firewall#
show run int e1/2
!
interface Ethernet1/2
nameif INSIDE
cts manual
propagate sgt preserve-untag
policy static sgt disabled trusted
security-level 0
ip address 192.168.0.1 255.255.255.0
ipv6 address
fc00:1:1:1::1/64

```
ipv6 enable
firewall#
show run int e1/3.202
!
interface Ethernet1/3.202
vlan 202
nameif OUTSIDE
cts manual
propagate sgt preserve-untag
policy static sgt disabled trusted
security-level 0
ip address 192.168.103.96 255.255.255.0
ipv6 address
fc00:1:1:2::1/64
ipv6 enable
```

La configuration du périphérique en amont révèle l'erreur de configuration :

```
<#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
```

Captures - Scénario fonctionnel

La modification du masque de sous-réseau (de /48 à /64) a résolu le problème. Il s'agit de la capture CAPI dans le scénario fonctionnel.

No.	Time	Source	Destination	colorength Info	
1	2019-10-24 15:17:20.677775	fc00:1:1:1::100	ff02::1:ff00:1	Pve 86 Neighbor Solic:	itation for fc00:1:1:1::1 from 4c:4e:35:fc:fc:d8
2	2019-10-24 15:17:20.677989	fc00:1:1:1::1	fc00:1:1:1::100	Pv6 🕗 86 Neighbor Adver	tisement fc00:1:1:1::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:ae
3	2019-10-24 15:17:20.678401	fc00:1:1:1::100	fc00:1:1:2::2	Pv6 114 Echo (ping) ree	quest id=0x097e, seq=0, hop limit=64 (no response found!)
4	2019-10-24 15:17:22.674281	fc00:1:1:1::100	fc00:1:1:2::2	Pv6 114 Echo (ping) ree	<pre>quest id=0x097e, seq=1, hop limit=64 (no response found!)</pre>
5	2019-10-24 15:17:24.674403	fc00:1:1:1::100	fc00:1:1:2::2	Pv6 114 Echo (ping) ree	quest id=0x097e, seq=2, hop limit=64 (reply in 6)
6	2019-10-24 15:17:24.674815	fc00:1:1:2::2	fc00:1:1:1::100	Pv6 114 Echo (ping) rep	ply id=0x097e, seq=2, hop limit=64 (request in 5)
7	2019-10-24 15:17:24.675242	fc00:1:1:1::100	fc00:1:1:2::2	Pv6 114 Echo (ping) ree	quest id=0x097e, seq=3, hop limit=64 (reply in 8)
8	2019-10-24 15:17:24.675731	fc00:1:1:2::2	fc00:1:1:1::100	Pv6 114 Echo (ping) rep	ply id=0x097e, seq=3, hop limit=64 (request in 7)
9	2019-10-24 15:17:24.676356	fc00:1:1:1::100	fc00:1:1:2::2	Pv6 114 Echo (ping) ree	quest id=0x097e, seq=4, hop limit=64 (reply in 10)
10	2019-10-24 15:17:24.676753	fc00:1:1:2::2	fc00:1:1:1::100	Pv6 114 Echo (ping) rep	ply id=0x097e, seq=4, hop limit=64 (request in 9)

Point clé :

1. Le routeur envoie un message de sollicitation de voisin IPv6 qui demande l'adresse MAC du

périphérique en amont (IP fc00:1:1:1::1).

- 2. Le pare-feu répond avec une annonce de voisin IPv6.
- 3. Le routeur envoie des requêtes d'écho ICMP et obtient des réponses d'écho.

Contenu CAPO :

1	io. Time	Source	Destination	Protoco	ingth Info
I	1 2019-10-24 15:17:20.678645	fe80::2be:75ff:fe	ff02::1:ff00:2	ICM	90 Neighbor Solicitation for fc00:1:1:2::2 from 00:be:75:f6:1d:8e
I	2 2019-10-24 15:17:20.681818	fc00:1:1:2::2	fe80::2be:75ff:fe	IC Z	90 Neighbor Advertisement fc00:1:1:2::2 (rtr, sol, ovr) is at 4c:4e:35:fc:fc:d8
I	3 2019-10-24 15:17:22.674342	fc00:1:1:1::100	fc00:1:1:2::2	ICMPN 3	118 Echo (ping) request id=0x097e, seq=1, hop limit=64 (reply in 6)
I	4 2019-10-24 15:17:22.677943	fc00:1:1:2::2	ff02::1:ff00:1	1(4)	90 Neighbor Solicitation for fc00:1:1:2::1 from 4c:4e:35:fc:fc:d8
I	5 2019-10-24 15:17:22.678096	fc00:1:1:2::1	fc00:1:1:2::2	ICMPV6 5	90 Neighbor Advertisement fc00:1:1:2::1 (rtr, sol, ovr) is at 00:be:75:f6:1d:8e
I	6 2019-10-24 15:17:22.678462	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118 Echo (ping) reply id=0x097e, seq=1, hop limit=64 (request in 3)
I	7 2019-10-24 15:17:24.674449	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118 Echo (ping) request id=0x097e, seq=2, hop limit=64 (reply in 8)
I	8 2019-10-24 15:17:24.674785	fc00:1:1:2::2	fc00:1:1:1::100	ICMPve	118 Echo (ping) reply id=0x097e, seq=2, hop limit=64 (request in 7)
I	9 2019-10-24 15:17:24.675395	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118 Echo (ping) request id=0x097e, seq=3, hop limit=64 (reply in 10)
I	10 2019-10-24 15:17:24.675700	fc00:1:1:2::2	fc00:1:1:1::100	ICMPv6	118 Echo (ping) reply id=0x097e, seq=3, hop limit=64 (request in 9)
I	11 2019-10-24 15:17:24.676448	fc00:1:1:1::100	fc00:1:1:2::2	ICMPv6	118 Echo (ping) request id=0x097e, seq=4, hop limit=64 (reply in 12)
	12 2019-10-24 15:17:24 676738	fc00:1:1:2::2	fc00.1.1.1.1.100	ICMPv6	118 Echo (ning) reply id=0x007e seq=4 hop limit=64 (request in 11)

Principaux points :

- 1. Le pare-feu envoie un message de sollicitation de voisin IPv6 qui demande l'adresse MAC du périphérique en amont (IP fc00:1:1:2::2).
- 2. Le pare-feu répond avec une annonce de voisin IPv6.
- 3. Le pare-feu envoie une requête d'écho ICMP.
- 4. Le routeur envoie un message de sollicitation de voisin IPv6 qui demande l'adresse MAC du périphérique en aval (IP fc00:1:1:1:1).
- 5. Le pare-feu répond avec une annonce de voisin IPv6.
- 6. Le pare-feu envoie des requêtes d'écho ICMP et obtient des réponses d'écho.

Cas 12. Problème de connectivité intermittent (empoisonnement ARP)

Description du problème : les hôtes internes (192.168.0.x/24) présentent des problèmes de connectivité intermittents avec les hôtes du même sous-réseau

Cette image présente la topologie :



Flux affecté :

Adresse IP source : 192.168.0.x/24

Adresse IP d'expédition : 192.168.0.x/24

Protocole : tout

Le cache ARP d'un hôte interne semble être empoisonné :

C:\Windows\system32\cmd.ex	e		
C:\Users\mzafeiro1>arp	-a		^
Interface: 192.168.0.55 Internet Address 192.168.0.1 192.168.0.22 192.168.0.23 192.168.0.24 192.168.0.25 192.168.0.26 192.168.0.27 192.168.0.28 192.168.0.29 192.168.0.30 192.168.0.30 192.168.0.255 224.0.0.255 224.0.0.251 224.0.0.251 224.0.0.252 239.255.255.250	$0xb$ Physical Address 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae 00-be-75-f6-1d-ae ff-ff-ff-ff-ff-ff 01-00-5e-00-00-16 01-00-5e-00-00-fc 01-00-5e-7f-ff-fa	Type dynamic dynamic dynamic dynamic dynamic dynamic dynamic dynamic dynamic static static static static static static	
C:\Users\mzafeiro1>_			•

Analyse de capture

Activer une capture sur le moteur LINA FTD

Cette capture ne capture que les paquets ARP sur l'interface INSIDE :

<#root>

firepower#

capture CAPI_ARP interface INSIDE ethernet-type arp



Captures - Scénario non fonctionnel :

La capture sur l'interface INSIDE du pare-feu contient.

	arp.dst.proto_ipv4 == 192.168.0.0/24) && !(ar	(arp.dst.proto_ipv4 == 192.168.0.0/24) && !(arp.src.proto_ipv4 == 192.168.0.1)										
N	No. Time	Source	Destination	Protocol rength Info	Info							
	4 2019-10-25 10:01:55.17957	1Vmware_2c:9b:a7	Broadcast	ARP U 60 Who has 192.168.0.23? Tell 192.168.0.55	Who has 192.168.0.23? Tell 192.1	.55						
	5 2019-10-25 10:01:55.17969	2 Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP 🛛 👗 42 192.168.0.23 is at 00:be:75:f6:1d:ae 🛃	192.168.0.23 is at 00:be:75:f6:1	(2)						
	35 2019-10-25 10:02:13.05039	<pre>7 Vmware_2c:9b:a7</pre>	Broadcast	ARP 60 Who has 192.168.0.24? Tell 192.168.0.55	Who has 192.168.0.24? Tell 192.1	.55						
	36 2019-10-25 10:02:13.05048	<pre>8 Cisco_f6:1d:ae</pre>	Vmware_2c:9b:a7	ARP 42 192.168.0.24 is at 00:be:75:f6:1d:ae [2]	192.168.0.24 is at 00:be:75:f6:1	2						
	47 2019-10-25 10:02:19.28468	3 Vmware_2c:9b:a7	Broadcast	ARP 60 Who has 192.168.0.25? Tell 192.168.0.55	Who has 192.168.0.25? Tell 192.1	.55						
	48 2019-10-25 10:02:19.28477	5 Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP 🖳 42 192.168.0.25 is at 00:be:75:f6:1d:ae 💋	192.168.0.25 is at 00:be:75:f6:1	2						
	61 2019-10-25 10:02:25.77982	1 Vmware_2c:9b:a7	Broadcast	ARP 60 Who has 192.168.0.26? Tell 192.168.0.55	Who has 192.168.0.26? Tell 192.1	.55						
	62 2019-10-25 10:02:25.77991	<pre>2 Cisco_f6:1d:ae</pre>	Vmware_2c:9b:a7	ARP 42 192.168.0.26 is at 00:be:75:f6:1d:ae [2]	192.168.0.26 is at 00:be:75:f6:1	2						
	76 2019-10-25 10:02:31.97817	5 Vmware_2c:9b:a7	Broadcast	ARP 60 Who has 192.168.0.27? Tell 192.168.0.55	Who has 192.168.0.27? Tell 192.1	.55						
	77 2019-10-25 10:02:31.97825	1 Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP 42 192.168.0.27 is at 00:be:75:f6:1d:ae	192.168.0.27 is at 00:be:75:f6:1	2						
	97 2019-10-25 10:02:38.66651	5 Vmware_2c:9b:a7	Broadcast	ARP 60 Who has 192.168.0.28? Tell 192.168.0.55	Who has 192.168.0.28? Tell 192.1	.55						
	98 2019-10-25 10:02:38.66660	6 Cisco_f6:1d:ae	Vmware_2c:9b:a7	ARP 🛛 💆 42 192.168.0.28 is at 00:be:75:f6:1d:ae 💋	192.168.0.28 is at 00:be:75:f6:1	2						
	121 2019-10-25 10:02:47.38407	4 Vmware_2c:9b:a7	Broadcast	ARP 60 Who has 192.168.0.29? Tell 192.168.0.55	Who has 192.168.0.29? Tell 192.1	.55						
	122 2019-10-25 10:02:47.38415	<pre>0 Cisco_f6:1d:ae</pre>	Vmware_2c:9b:a7	ARP 42 192.168.0.29 is at 00:be:75:f6:1d:ae [2]	192.168.0.29 is at 00:be:75:f6:1	2						
	137 2019-10-25 10:02:53.53999	5 Vmware_2c:9b:a7	Broadcast	ARP 60 Who has 192.168.0.30? Tell 192.168.0.55	Who has 192.168.0.30? Tell 192.1	.55						
	138 2019-10-25 10:02:53.54008	<pre>7 Cisco_f6:1d:ae</pre>	Vmware_2c:9b:a7	ARP 🛛 🕊 42 192.168.0.30 is at 00:be:75:f6:1d:ae 💋	192.168.0.30 is at 00:be:75:f6:1	2						

Principaux points :

- 1. Le pare-feu reçoit diverses requêtes ARP pour les adresses IP du réseau 192.168.0.x/24
- 2. Le pare-feu répond à tous ces paquets (proxy-ARP) avec sa propre adresse MAC

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Vérifiez la configuration NAT.

En ce qui concerne la configuration NAT, il y a des cas où le mot clé no-proxy-arp peut empêcher le comportement précédent :

<#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

Action 2. Désactivez la fonctionnalité proxy-arp sur l'interface du pare-feu.

Si le mot clé « no-proxy-arp » ne résout pas le problème, essayez de désactiver le proxy ARP sur l'interface elle-même. Dans le cas de FTD, au moment de la rédaction de ce document, vous devez utiliser FlexConfig et déployer la commande (spécifiez le nom d'interface approprié).

sysopt noproxyarp INSIDE

Cas 13 . Identifier les identificateurs d'objet SNMP (OID) qui provoquent des erreurs de CPU

Ce cas montre comment certains OID SNMP pour l'interrogation de la mémoire ont été identifiés comme la cause principale des erreurs de CPU (problème de performances) sur la base de l'analyse des captures de paquets SNMP version 3 (SNMPv3).

Description du problème : les dépassements sur les interfaces de données augmentent constamment. D'autres recherches ont révélé qu'il y a aussi des erreurs de CPU (causées par le processus SNMP) qui sont la cause première des dépassements d'interface.

L'étape suivante du processus de dépannage consistait à identifier la cause première des erreurs de CPU provoquées par le processus SNMP et, en particulier, à réduire l'étendue du problème pour identifier les identificateurs d'objet SNMP (OID) qui, lorsqu'ils sont interrogés, peuvent potentiellement entraîner des erreurs de CPU.

Actuellement, le moteur FTD LINA ne fournit pas de commande « show » pour les OID SNMP qui sont interrogés en temps réel.

La liste des OID SNMP pour l'interrogation peut être récupérée à partir de l'outil de surveillance SNMP, cependant, dans ce cas, il y avait ces facteurs préventifs :

- L'administrateur FTD n'a pas eu accès à l'outil de surveillance SNMP
- SNMP version 3 avec authentification et cryptage des données pour la confidentialité a été configuré sur FTD

Analyse de capture

Comme l'administrateur FTD disposait des informations d'identification pour l'authentification SNMP version 3 et le cryptage des données, ce plan d'action a été proposé :

- 1. Capture des paquets SNMP
- 2. Enregistrez les captures et utilisez les préférences de protocole SNMP de Wireshark pour

spécifier les informations d'identification SNMP version 3 afin de déchiffrer les paquets SNMP version 3. Les captures décryptées sont utilisées pour l'analyse et la récupération des OID SNMP

Configurez les captures de paquets SNMP sur l'interface utilisée dans la configuration d'hôte snmp-server :

<#root>

firepower#

show run snmp-server | include host

snmp-server host management 192.168.10.10 version 3 netmonv3

firepower#

show ip address management

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

firepower#

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

firepower#

show capture capsnmp

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

9512

bytes] match udp host 192.168.10.10 host 192.168.5.254 eq snmp

No.		Time	Protocol	Source	Source Port	Destination Port	Destination	Length	Info
-	1	0.000	SNMP	192.168.10.10	65484	161	192.168.5.254	100	getBulkRequest
	2	0.000	SNMP	192.168.5.254	161	65484	192.168.10.10	167	report 1.3.6.1.6.3.15.1.1.4.0
	3	0.176	SNMP	192.168.10.10	65484	161	192.168.5.254	197 2	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
i i	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
L	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
<									>
	<[Des	tination	Host: 19	2.168.5.254]>					
	<[Sou	ince on De	estinatio	n Host: 192.168.5.	254]>				
> Use	r Dat	tagnam Pro	otocol, S	Grc Port: 65484, D:	st Port: 10	51			
∨ Sim	ple M	letwork M	anagement	Protocol					
	msgVe	rsion: sr	nmpv3 (3)						
> 1	msgG1	obalData							
> 1	msgAu	thoritati	iveEngine	ID: 80000009fe1c6d	lad4930a00e	f1fec2301621	a4158bfc1f40		
	msgAu	thoritati	iveEngine	Boots: 0					
	msgAu	thoritati	iveEngine	Time: 0					
	msgUs	erName: r	netmonv3						
	msgAu	thenticat	tionParam	eters: ff5176f5973	c30b62ffc1	168			
	msgPr	ivacyPara	ameters:	000040e100003196					
~	∨ msgData: encryptedPDU (1)								
E	B encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703_								
_									

Principaux points :

- 1. Adresses/ports source et de destination SNMP.
- 2. La PDU du protocole SNMP n'a pas pu être décodée car privKey est inconnu de Wireshark.
- 3. La valeur de la primitive encryptéePDU.

Actions recommandées

Les actions répertoriées dans cette section ont pour objectif de réduire davantage le problème.

Action 1. Déchiffrez les captures SNMP.

Enregistrez les captures et modifiez les préférences du protocole SNMP Wireshark pour spécifier les informations d'identification SNMP version 3 permettant de déchiffrer les paquets.

<#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
IIIIII
64 packets copied in 0.40 secs

Ouvrez le fichier de capture sur Wireshark, sélectionnez un paquet SNMP et naviguez jusqu'à Protocol Preferences > Users Table, comme indiqué dans l'image :

No.	1 2	Time 0.000 0.000 0.176	Protocol SNMP SNMP SNMP	Source 192.168.10.10 192.168.5.254 192.168.10.10	Source Port 65484 161 65484	Destination Port 161 65484	Destination 192.168.5.2 192.168.10.1	Length 54 100 10 167		bró getBulkRequest report 1.3.6.1.6.3.15.1.1.4.0 encryntedPDU: privKey Unknown	
	4 5 6 7 8 9 10 11 12 13 14 15 16 17	0.176 0.325 0.326 0.490 0.675 0.767 0.945 0.946 1.133 1.134 1.317 1.318 17.595	SIMP SIMP	192.168.5.254 192.168.5.254 192.168.5.254 192.168.5.254 192.168.5.254 192.168.5.254 192.168.8.0.10 192.168.8.0.10 192.168.5.254 192.168.10.10 192.168.5.254 192.168.10.10	161 65484 161 65484 161 65484 161 65484 161 65484 161 65484 161 65484	Mark/Unma Ignore/Unig Set/Unset T Time Shift Packet Com Edit Resolve Apply as Fil Prepare a Fi Conversatio Colorize Co SCTP Follow	rk Packet inner Packet ime Reference ment id Name ter iter n Filter nversation	Ctrl+M Ctrl+D Ctrl+T Ctrl+Shift+T Ctrl+Alt+C		encryptedPDU: privKey Unknown encryptedPDU: privKey Unknown	
	18 19	17.595 17.749	SNMP	192.168.5.254 192.168.10.10	161 62008	Protocol Pre	ferences			report 1.3.6.1.6.3.15.1.1.4.0 Open Simple Network Management Protocol preferences	
	20 21 22 23	17.749 17.898 17.899 18.094	SNMP SNMP SNMP SNMP	192.168.5.254 192.168.10.10 192.168.5.254 192.168.10.10	161 62008 161 62008	Decode As. Show Packe	t in New Window 192.100.10.1	54 205	~ ~ ~	Show SNMP OID in info column Reassemble SNMP-over-TCP messages spanning multiple TCP segments Display dissected variables inside SNMP tree	
	24	18.094	SNMP	192.168.5.254	161 62008	62008 161	192.168.10.1	10 560 54 205		Users Table Enterprise Specific Tran Types	
< > U ~ S	<pre><[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)</pre>							- 207		SMP UDP port: 161 SNMP TCP port: 161 Disable SNMP	>

Dans le tableau SNMP Users, le nom d'utilisateur, le modèle d'authentification, le mot de passe d'authentification, le protocole de confidentialité et le mot de passe de confidentialité SNMP version 3 ont été spécifiés (les informations d'identification réelles ne sont pas indiquées cidessous) :

🚄 SNMP U	sers					?	×
Engine ID	Username	Authentication model	Password	Privacy protocol	Privacy password		
		MD5		DES			
+ -	• ~ ~		<u>C: Use</u>	rs\igasimov\AppData\	Roaming Wireshark profiles Pro	ofile1 snmp	users
				OK	Copy from 👻 Cancel	Hel	2

Une fois que les paramètres SNMP Users ont été appliqués, Wireshark a montré les PDU SNMP déchiffrées :

1 0.000 SNPP 192.168.10.10 65484 161 192.168.5.254 100 getBulkRequest 2 0.000 SNPP 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 SNPP 192.168.5.254 161 65484 192.168.5.254 197 getBulkRequest 4 0.176 SNPP 192.168.3.254 161 192.168.10.10 167 report 1.3.6.1.6.3.15.1.1.4.0 5 0.325 SNPP 192.168.3.254 161 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 6 0.326 SNPP 192.168.5.254 161 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 6 0.326 SNPP 192.168.10.10 65484 161 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 1.3.6.1.4.1.9.9.221.1 7 0.490 SNPP 192.168.10.10 65484 161 192.168.5.254 205 <th>.1.2 1.3.6.1.4.1.9.9.221.1.1 .1.2 1.3.6.1.4.1.9.9.221.1.1</th>	.1.2 1.3.6.1.4.1.9.9.221.1.1 .1.2 1.3.6.1.4.1.9.9.221.1.1								
2 0.000 SNMP 192.168.5.254 161 65484 192.168.1.0.10 167 report 1.3.6.1.6.3.15.1.1.4.0 3 0.176 SNMP 192.168.5.254 161 192.168.5.254 197 getBulkRequest 1.3.6.1.6.3.15.1.1.2.0 4 0.176 SNMP 192.168.5.254 161 65484 192.168.10.10 157 5 0.325 SNMP 192.168.10.10 65484 161 192.168.5.254 199 getBulkRequest 1.3.6.1.6.3.15.1.1.2.0 6 0.325 SNMP 192.168.5.254 161 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 6 0.326 SNMP 192.168.10.20 65484 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 7 0.490 SNMP 192.168.10.10 65484 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 7 0.490 SNMP 192.168.5.10.10 65484 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9									
3 0.176 SWP 192.168.10.10 65484 161 192.168.254 197 getBulkRequest 1.3.6.1.4.1.9.9.221.1 4 0.176 SWP 192.168.254 161 192.168.254 197 report 1.3.6.1.4.1.9.9.221.1 5 0.325 SWP 192.168.10.10 65484 161 192.168.5.254 197 report 1.3.6.1.4.1.9.9.221.1 6 0.326 SWP 192.168.10.26 65484 161 192.168.10.10 678 getBulkRequest 1.3.6.1.4.1.9.9.221.1 7 0.490 SWP 192.168.10.10 65484 161 192.168.5.254 getBulkRequest 1.3.6.1.4.1.9.9.221.1 7 0.490 SWP 192.168.10.10 65484 161 192.168.5.254 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.1.1.3.1.8	.1.2 1.3.6.1.4.1.9.9.221.1.1 .1.2 1.3.6.1.4.1.9.9.221.1.1								
4 0.176 SNMP 192.168.5.254 161 65484 192.168.10.10 192 report 1.3.6.1.6.3.15.1.1.2.0 5 0.325 SNMP 192.168.10.10 65484 161 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 6 0.326 SNMP 192.168.5.254 161 65484 192.168.10.10 678.0 get-response 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8 7 0.490 SNMP 192.168.10.10 65484 161 192.168.5.254 205 getbulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8	.1.2 1.3.6.1.4.1.9.9.221.1.1 .1.2 1.3.6.1.4.1.9.9.221.1.1								
5 0.325 SWP 192.168.10.10 65484 161 192.168.5.254 199 getBulkRequest 1.3.6.1.4.1.9.9.221.1 6 0.326 SWP 192.168.5.254 161 65484 192.168.5.254 get-response 1.3.6.1.4.1.9.9.221.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2. 7 0.490 SWP 192.168.10.10 65484 161 192.168.5.254 295 get-response 1.3.6.1.4.1.9.9.221.1.1.1.3.1.8	.1.2 1.3.6.1.4.1.9.9.221.1.1								
6 0.326 SNMP 192.168.5.254 161 65484 192.168.10.10 678 0 get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2 7 0.490 SNMP 192.168.10.10 65484 161 192.168.5.254 205 n getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8	2.1.2 1.3.6.1.4.1.9.9.221.1.1 .1.2 1.3.6.1.4.1.9.9.221.1.1								
7 0.490 SNMP 192.168.10.10 65484 161 192.168.5.254 205 🞧 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8	.1.2 1.3.6.1.4.1.9.9.221.1.1								
	.1.2 1.3.6.1.4.1.9.9.221.1.1								
8 0.490 SNMP 192.168.5.254 161 65484 192.168.10.10 560 👰 get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.5									
9 0.675 SNMP 192.168.10.10 65484 161 192.168.5.254 205 1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8									
10 0.767 SNMP 192.168.5.254 161 65484 192.168.10.10 610 Ø get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.7.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.7.	.1.2 1.3.6.1.4.1.9.9.221.1.1								
11 0.945 SNMP 192.168.10.10 65484 161 192.168.5.254 205 🕦 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.8.1.8									
12 0.946 SNMP 192.168.5.254 161 65484 192.168.10.10 584 👸 get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	17.1.2 1.3.6.1.4.1.9.9.221.1								
13 1.133 SNMP 192.168.10.10 65484 161 192.168.5.254 205 1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.18.1.8									
14 1.134 SNMP 192.168.5.254 161 65484 192.168.10.10 588 get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.1.9.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.	19.1.2 1.3.6.1.4.1.9.9.221.1								
15 1.317 SNMP 192.168.10.10 65484 161 192.168.5.254 205 1 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.20.1.8									
□ 16 1.318 SNMP 192.168.5.254 161 65484 192.168.10.10 513 0 get-response 1.3.6.1.4.1.9.9.392.1.1.1.0 1.3.6.1.4.1.9.9.392.1.1.2.0 1.3.6.1	4.1.9.9.392.1.1.3.0 1.3.6.1								
17 17.595 SNMP 192.168.10.10 62008 161 192.168.5.254 100 getBulkRequest									
18 17.595 SNMP 192.168.5.254 161 62008 192.168.10.10 167 report 1.3.6.1.6.3.15.1.1.4.0									
19 17.749 SNMP 192.168.10.10 62008 161 192.168.5.254 197 🕕 getBulkRequest 1.3.6.1.4.1.9.9.221.1									
20 17.749 SIMP 192.168.5.254 161 62008 192.168.10.10 192 report 1.3.6.1.6.3.15.1.1.2.0									
21 17.898 SNMP 192.168.10.10 62008 161 192.168.5.254 199 🕕 getBulkRequest 1.3.6.1.4.1.9.9.221.1									
22 17.899 SIMP 192.168.5.254 161 62008 192.168.10.10 678 (2) get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.2.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.2	.1.2 1.3.6.1.4.1.9.9.221.1.1								
23 18.094 SIMP 192.168.10.10 62008 161 192.168.5.254 205 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.3.1.8									
24 18.094 SNMP 192.168.5.254 161 62008 192.168.10.10 560 get-response 1.3.6.1.4.1.9.9.221.1.1.1.1.5.1.1 1.3.6.1.4.1.9.9.221.1.1.1.1.5	.1.2 1.3.6.1.4.1.9.9.221.1.1								
25 18.290 SNMP 192.168.10.10 62008 161 192.168.5.254 205 getBulkRequest 1.3.6.1.4.1.9.9.221.1.1.1.1.6.1.8									
	3								
✓ msgData: encryptedPDU (1)									
encryptedPDU: 879a16d23633400a0391c5280d226e0cec844d87101ba703									
✓ Decrypted ScopedPDU: 303b04198000009fe1c6dad4930a00ef1fec2301621a415									
> 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)									
(b)gect Name: 1.3.6.1.4.1.9.9.221.1 (1so.3.6.1.4.1.9.9.221.1)									
Value (Null)									

Principaux points :

- 1. Les outils de surveillance SNMP ont utilisé SNMP getBulkRequest pour interroger et parcourir l'OID parent 1.3.6.1.4.1.9.9.221.1 et les OID associés.
- 2. Le FTD a répondu à chaque getBulkRequest avec get-response qui contient des OID associés à 1.3.6.1.4.1.9.9.221.1.

Action 2. Identifiez les OID SNMP.

<u>SNMP Object Navigator</u> a montré que l'OID 1.3.6.1.4.1.9.9.221.1 appartient à la base d'informations de gestion (MIB) nommée CISCO-ENHANCED-MEMPOOL-MIB, comme indiqué dans l'image :

Tools & Resources SNMP Object Na	avigator				
HOME SUPPORT TOOLS & RESOURCES SNMP Object Navigator	TRANSLATE/BROWSE Translate Browse The Translate OID into object name Enter OID or object name:	SEARCH I Object Tree ne or object name 3.6.1.4.1.9.9.22 Iranslate	DOWNLOAD MIBS into OID to receive ot 1.1	MIB SUPPORT - SW oject details examples - OID: 1.3.6.1.4.1.9.9.27 Object Name: ifIndex	Help [-] Feedback Related Tools Support Case Manager Cisco Community. MIB Locator
	Object Information Object cer OID 1.3 MIB CIS OID Tree You are currently viewing you	-			
	. <u>iso.(1).</u> . <u>org.(3).</u> . <u>dod.(6).</u> . <u>int</u> <u>ciscoMgmt (9)</u> _ <u>+</u> <u>ciscoTcpMIB.(6)</u> 				

Pour afficher les OID dans un format lisible par l'utilisateur dans Wireshark :

1. Téléchargez la MIB CISCO-ENHANCED-MEMPOOL-MIB et ses dépendances, comme illustré dans l'image :

HOME	TRANSLATE/BROWSE	SEARCH	DOWNLOAD MIBS	MIB SUPPORT - SW	Help [-] Feedback
UPPORT					Related Tools
OOLS & RESOURCES SNMP Object Navigator	View MIB dependencies and	Support Case Manager Cisco Community MIB Locator			
	Step 1. Select a MIB name b CISCO-ENHANCED-ME	w typing or screen MPOOL-MIB	olling and then select a fi	unction in step 2 and click Submit	
	List matching MIBs				
	A100-R1-MIB			A	
	ACTONA-ACTASTOR-M	IB			
	ADMIN-AUTH-STATS-M	IB			
	ADSL-DMT-LINE-MIB				
	ADSL-LINE-MIB				
	ADSL-TC-MIB				
	ADSI 2-LINE-MIR			*	

HOME	TRANSLATE/BROWSE SEARCH	DOWNLOAD MIE	MIB SUF	PPORT - SW	Help [+] Feedback
SUPPORT					Related Tools
OOLS & RESOURCES					Support Case Manager
SNMP Object Navigator	CISCO-ENHANCED-MEMPOOL-MIB				Cisco Community MIB Locator
	View compiling dependencies for other MIE	S by clearing the page	ge and selecting	another MIB.	
	Compile the MIB				
	Before you can compile CISCO-ENHANCE below in the order listed.	D-MEMPOOL-MIB,	you need to com	pile the MIBs listed	
	Download all of these MIBs (Warning: does MIB below.	not include non-Cisc	co MIBs) or view	details about each	
	If you are using Internet Explorer click here				
	MIB Name	Version 1	Version 2	Dependencies	
	1. SNMPv2-SMI	Download	Download	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	-	
	9. RFC-1212	Non-Cisco MIB	Non-Cisco	1 · · ·	
	10. RFC-1215	Non-Cisco	Non-Cisco	I -	
	11. SNMPv2-TC-v1	Non-Cisco	Non-Cisco		
	12. CISCO-ENHANCED-	MIB	MIB		

2. Dans Wireshark dans la fenêtre Edit > Preferences > Name Resolution, l'option Enable OID Resolution est cochée. Dans la fenêtre SMI (MIB and PIB paths), spécifiez le dossier avec les MIB téléchargées et dans SMI (MIB and PIB modules). La MIB CISCO-ENHANCED-MEMPOOL est ajoutée automatiquement à la liste des modules :

The Protect I. The Protect I. State State									
4 0.176 SHP Winshuft-Arbetenees 7 X 5 0.125 SHP Readw HKC defenses 7 X 7 0.600 SHP Readw HKC defenses 1.4.1.9.9.221.1.1 10 0.605 SHP Feedow Hanges Tames 1.4.1.9.9.221.1.1 11 0.605 SHP Feedow Hanges Tames 1.4.1.9.9.221.1.1 12 0.605 SHP Feedow Hanges Tames 1.4.1.9.9.221.1.1 13 0.636 SHP Feedow Hanges Tames 1.4.1.9.9.221.1.1 14 1.133 SHP Periodicities 6.1.4.1.9.9.221.1.1 14 1.133 SHP Septem SHP Feedow Hanges Tames 6.1.4.1.9.9.221.1.1 15 1.317 SHP Septem	No.	Time	Protocol	Source S	ource Port Destination Port Destination Length	Info			- · · · · ·
5 0.325 SHPP 7 0.408 SHPP 9 0.675 SHPP 9 0.675 SHPP 10 0.775 SHPP 11 0.945 SHPP 12 0.946 SHPP 9 0.675 SHPP 12 0.946 SHPP 13 0.775 SHPP 12 0.946 SHPP 13 0.7749 SHPP 14 1.135 SHPP 14 1.135 SHPP 15 1.7749 SHPP 12 1.749 SHPP 13 1.755 SHPP 15 1.7749 SHPP 16 1.7749 SHPP 16 1.7749 SHPP 16 1.7749 SHPP 16 1.7749 SHPP 15 1.7749 SHPP 15 1.7749 SHPP 15 1.7749 SHPP 15 1.7749 SHPP 15 1.7749 SHPP 16 1.7749 SHPP 16 1.7749 SHPP 16 1.14,1,9,9,221,1,1 14,1,9,9,221,1 14,1,9,9,	4	0.176	SNMP	Wireshark - Preference	25	?	\times	MI Paths ? X	
6 0.26 SNP V Appealance Columns Columns	5	0.325	SNMP						
7 0.480 SHP Columns Instant stants (Sp) data starts in solver starts (Sp) data starts	6	0.326	SNMP	✓ Appearance	Resolve MAC addresses		^	Directory path	.1.4.1.9.9.221.1.1
8 0.400 SNPP 9 0.675 SNPP 10 0.767 SNPP 11 0.975 SNPP 12 0.966 SNPP 13 1.133 SNPP 13 1.133 SNPP 14 1.134 SNPP 15 1.317 SNPP 15 1.317 SNPP 15 1.317 SNPP 15 1.318 SNPP 19 17 21:55 SNPP 19 17 21:52 SNPP 19 21 17.809 SNPP 22 11.809 SNPP 22 11.809 SNPP 22 11.809 SNPP 22 11.809 SNPP 23 11.809 SNPP 23 11.809 SNPP 24 11.804 SNPP 24 11.804 SNPP 25 SNPP 10 Filter Balance and Tabase detorms 16 Cm 10 Filter Additional State and Tabase detorms 10 Filter Additional State Additional Sta	7	0.490	SNMP	Columns	Resolve transport names			C:// kers/Administrator/Downloads/SNMPMIRS	
9 0.675 SNPP Lupurt Image: Network Network Network P/ Solutions 10 0.775 SNPP Capture Use capture diverses resolution 11 0.945 SNPP SNPP SNPP SNPP 12 0.946 SNPP Name: Resolution Image: Resolution Image: Resolution 13 1.133 SNPP Name: Resolution Image: Resolution	8	0.490	SNMP	Font and Colors				C/ OKT # Partimistrator/ Downloads # Strain Midd	.1.4.1.9.9.221.1.1
10 0.767 SMP Capture If we raphed toto pack due for dadees readotion 11 0.965 SMP If we raphed toto pack due for dadees readotion .1.4.1.9.9.221.1.1 12 0.966 SMP Nime Readotion .6.1.4.1.9.9.221.1.1 13 1.133 SMP Nime Readotion .6.1.4.1.9.9.221.1.1 14 1.134 SMP Potocois RSA Keys .6.1.4.1.9.9.221.1.1 15 1.317 SMP Sati Via us fill modes .6.1.4.1.9.9.221.1.1 .6.1.4.1.9.9.221.1.1 15 1.318 SMP SMP SMP modes .6.1.4.1.9.9.221.1.1 .6.1.4.1.9.9.221.1.1 16 1.318 SMP SMP modes .6.1.4.1.9.9.221.1.1 .6.1.4.1.9.9.221.1.1 17 17.575 SMP .3.1.6.19 modes .6.1.4.1.9.9.221.1.1 .6.1.4.1.9.9.221.1.1 21 1.4.1.9.9.221.1.1 .6.1.4.1.9.9.221.1.1.1 .6.1.4.1.9.9.221.1.1.1 .6.1.4.1.9.9.221.1.1.1 21 1.5.68 SMP	9	0.675	SNMP	Layout	Resolve network (IP) addresses				
<pre>11 0.945 SMP 12 0.946 SMP 13 1.133 SMP 14 1.134 SMP 15 1.317 SMP 15 1.317 SMP 15 1.317 SMP 16 1.318 SMP 16 1.318 SMP 17 17.595 SMP 19 17.749 SMP 20 17.749 SMP 20 17.749 SMP 20 17.749 SMP 21 17.988 SMP 22 17.989 SMP 23 18.964 SMP 23 18.964 SMP 24 18.964 SMP 24 18.964 SMP 25 11trent Protocol Version 4, 57:192.168.01.0, Dst: a2:4c:66:00:00:20) 24 18.964 SMP 24 18.964 SMP 25 11trent Protocol Version 4, 57:192.168.01.0, Dst: a2:4c:66:00:00:20) 24 18.964 SMP 25 11trent Protocol Version 4, 57:192.168.01.0, Dst: a2:4c:66:00:00:20) 25 Simple Network Management Protocol 25 Simple Network Management Protocol 25 Simple Network Management Protocol</pre>	10	0.767	SNMP	Capture	Use captured DNS packet data for address resolution				.1.4.1.9.9.221.1.1
12 0.966 SHPP 13 1.133 SHP 14 1.134 SHPP 16 1.313 SHP 16 1.318 SHPP 16 1.318 SHPP 16 1.318 SHPP 16 1.318 SHPP 20 17.729 SHPP 20 17.729 SHPP 20 17.729 SHPP 20 17.729 SHPP 20 17.729 SHPP 21 17.386 SHPP 22 17.386 SHPP 22 17.386 SHPP 22 17.386 SHPP 23 18.0924 SHPP 24 18.0924 SHPP 24 18.0924 SHPP 25 31(Maid database directories Edt. 24 18.0924 SHPP 25 31(Maid database directories Edt. 24 18.0924 SHPP 23 18.0924 SHPP 24 17.386 SHPP 25 31(Maid database directories Edt. 24 18.0924 SHPP 25 31(Cisco 33:fe:bf (00:12:7f:33:fe:bf), Dst: a2:4c:66:00:00:20) 1 Internet Tryotocol Version A, 55c: 192:168.10.10, Dst: 192:168:0.204 25 SHMP/COM/NIT/-MBB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXACE/VMB SHMP-FXA	11	0.945	SNMP	Expert	Use an external network name resolver				
<pre>1 1.33 StMP / StMP</pre>	12	0.946	SNMP	Filter Buttons	Maximum concurrent requests 500				.6.1.4.1.9.9.221.1
<pre>1 1.134 SH4P > Protocols SA Keys NUN DB SA Keys SS PCs = GetWAUN DB Sa Keys SS PCs = GetWAUN DB Seaked SS PCs = GetWater SS Herror SS Heror SS Herror SS He</pre>	13	1.133	SNMP	Name Resolution					
15 1.117 SINPP 16 1.118 SINPP 17 17.555 SINP 18 17.555 SINP 20 17.749 SINP 20 17.749 SINP 21 17.885 SINP 22 17.889 SINP 22 17.889 SINP 22 17.889 SINP 22 17.889 SINP 22 17.889 SINP 22 17.889 SINP 23 18.094 SINP 24 18.094 SINP 24 18.094 SINP 25 Trane 23: 285 bytes on kit Ethernet IP, Src: (size, 33:febf (00:12:7f:33:febf), Dst: a2:4c:66:00:00:20 (a2:4c:66:00:00:20) Ethernet IP, Src: (size, 33:febf (00:12:7f:33:febf), Dst: a2:4c:66:00:00:20 (a2:4c:66:00:00:20) Simple Network Management Protocol	14	1.134	SNMP	> Protocols	Only use the profile "hosts" file				.6.1.4.1.9.9.221.1
<pre>b f 1.318 SNAP 17 717.55 SNAP 19 17.745 SNAP 20 17.745 SNAP 21 17.89 SNAP 22 17.89 SNAP 23 18.094 SNAP 24 18.094 SNAP 54 Mill and PB) points Ede. 54 Mill and PB) points Ede. 55 Frame 23: 205 bytes on wit 5 F</pre>	15	1.317	SNMP	RSA Keys	Resolve VLAN IDs				
<pre>17 17.595 SHPP 18 17.595 SHPP 20 17.749 SHPP 20 17.749 SHPP 20 17.749 SHPP 22 17.898 SHPP 22 17.898 SHPP 22 17.899 SHPP 23 18.094 SHPP 24 18.094 SHPP 25 SHPP 26 SHPP 26 SHPP 27 SHPP 27 SHPP 28 SHPP 29 SHPP</pre>	16	1.318	SNMP	> Statistics	Resolve SS7 PCs				92.1.1.3.0 1.3.6.1
18 17.595 SHIP 19 17.749 SHIP 20 17.749 SHIP 21 17.895 SHIP 22 17.895 SHIP 23 18.094 SHIP 24 18.094 SHIP 24 18.094 SHIP 24 1.604 SHIP 25 205 bytes on via 26 1.61:12:7f:33:fe:16f), Dst: 192.168:5.254 20 Userbatagram Protocol, Src Port: 62008, Dst Port: 101 25 Simple Network Hanagement Protocol 26 Simple Network Hanagement Protocol 27 Simple Network Hanagement Protocol	L 17	17.595	SNMP	Advanced	C Enable OTD resolution				
<pre>i9 17.749 SteP 20 17.749 SteP 21 17.898 SteP 22 17.898 SteP 22 17.899 SteP 22 17.899 SteP 22 17.899 SteP 22 17.899 SteP 22 18.994 SteP 24 18.994 SteP 25 Frame 23: 205 bytes on wil > Efternet II, Src: (isco_33:fe:bf) (00:12:7f:33:fe:bf), Dst: a2:4c:66:00:00:20) > Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.5, 254 > User Datagram Protocol, Src Port: 62088, Dst Port: 161 > Simple Network Management Protocol > Simple Network Management Protocol</pre>	18	17.595	SNMP						
28 17.749 SNPP 21 17.898 SNPP 21 17.899 SNPP 23 18.994 SNPP 24 18.994 SNPP 25 Cox Cancel 24 18.994 SNPP 25 SNMP SMM Odules 2 11.4.1.9.9.221.1.1 3 Module name 2 11.4.1.9.9.221.1.1 3 Module name 3 Module name 9 Simple Network Management Protocol 3 Simple Network Management Protocol 3 Simple Network Management Protocol	19	17.749	SNMP		Suppress SMI errors				
21 17.898 SMP SMI (MB and PB) models Edt 22 17.898 SMP XM (MB and PB) models Edt 23 18.094 SMP XM (MB and PB) models Edt > Frame 23: 205 bytes on with XM (MB and PB) models Edt XM (MB and PB) models 1.4.1.9.9.221.1.1 > Ethernet II, Src: Cisco_33:feibf (00:12:7f:33:feibf), Dst: a2:4c:66:00:00:20 (a2:4c:66:00:00:20) XM (MUNTY-MB XMM - COMMUNTY-MB > User Datagram Protocol, Src Port: 62008, Dst Port: 161 SMM - KAMEWORK-MB SMM - KAMEWORK-MB > Simple Network Management Protocol SMM - Models XMM - KAMEWORK-MB SMM - NOTIFICATION-MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - NOTIFICATION-MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - NOTIFICATION-MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - KAMEWORK - MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - KAMEWORK - MB SMM - KAMEWORK-MB SMM - KAMEWORK-MB SMM - KAMEWORK - MB SMM - KAMEWORK - MB SMM - KAMEWORK - MB SMM - KAMEWORK - MB SMM - KAMEWORK - MB SMM - KAM	20	17.749	SNMP		SMI (MIB and PIB) paths Edit				
22 17.899 SMP SMI (MB and PB) modules Edt 23 18.994 SMP Maddind database directories Edt 24 18.994 SMP SMI Modules ? X 1.4.1.9.9.221.1.1 2 Frame 23: 205 bytes on wit DK Cancel Help 1.4.1.9.9.221.1.1 2 Frame 23: 205 bytes on wit SMI Modules ? X 1.4.1.9.9.221.1.1 2 Strapped Fortocol Version 4, Src: 192.168.19.10, Dst: a2:4c:66:00:00:20 (a2:4c:66:00:00:20) SMIP Module name IPV6-MIB 2 Simple Network Management Protocol SimPreconverting SNMP-MEDANIB SNMP-MOD-MIB SNMP-MEDANIB 3 Simple Network Management Protocol SimPreconverting SimPreconverting IPV6-MIB 3 SimPreconverting SimPreconverting SimPreconverting IPV6-MIB 3 SimPreconverting SimPreconverting SimPreconverting IPV6-MIB 3 SimPreconverting SimPreconverting IPV6-MIB SimPreconverting 3 SimPreconverting SimPreconverting IPV6-MIB IPV6-MIB 3 SimPreconverting	21	17.898	SNMP					I = № ∧ Y U6 <u>CillisersigasimoviApp0ataamingiWiresharkismi paths</u>	
23 18.094 SMMP 24 18.094 SMP 25 Frame 23: 205 bytes on wil Module name 11cterct Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.5.254 Module name 120 User Datagram Protocol, Src Port: 62088, Dst Port: 161 Simple Network Management Protocol Simple Network Management Protocol Simple Metwork.Management Protocol Simple Network Management Protocol Simple Metwork.Man	22	17.899	SNMP		SMI (MIB and PIB) modules Edit			OK Cancel Help	.1.4.1.9.9.221.1.1
74 18.094 SNMP C X 1.4.1.9.9.221.1.1* Y Y Y X <t< td=""><td>23</td><td>18,094</td><td>SNMP</td><td></td><td>MaxMind database directories Edit</td><td></td><td></td><td></td><td></td></t<>	23	18,094	SNMP		MaxMind database directories Edit				
C OK Cancel Help > Ertemet 11, Src: (15cc;03):fe:bf (00:12:7f:33:fe:bf), Dst: a2:4c:66:00:00:20 (a2:4c:66:00:00:20) > > Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.5.254 > User Datagram Protocol, Src Port: 62008, Dst Port: 161 > Simple Network Management Protocol Simple Network Management Protocol Viser Datagram Protocol, Src Port: 62008, Dst Port: 161 Simple Network Management Protocol	24	18,094	SNMP	< >			~	SMI Modules 2 X	1.4.1.9.9.221.1.1
> Frame 23: 205 bytes on with the constraint of the constraint	<				OX	Cancel Hele			>
> Tethernet II, Sr:: Cisco_33:fe:bf (00:12:7f:33:fe:bf), Dst: a2:4c:66:00:00:20) > Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.5.254 > User Datagram Protocol, Src Port: 62008, Dst Port: 161 Simple Network Management Protocol Simple Network Management Protocol Simple Network Management Protocol	> Ename 2	2 205 hr	tes on wi		UK UK	Cancel Heij	٢		
Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.5.254 User Datagram Protocol, Src Port: 62088, Dst Port: 161 Simple Network Management Protocol Window Management Protocol Simple Network Management Protocol	> Ethernet	TT Sec	· Cisco 3	3.fe.hf (00.12.7f.33	(a2.4c.66.00.00.20 (a2.4c.66.00.00	1.20)		Module name	
> User Datagram Protocol, Src Port: 62008, Dst Port: 161 Simple Network Management Protocol SNMP-RAMEWORK-KNIB SIMP-RAMEWORK-MIB SNMP-RAMEWORK-MIB SNMP-RAMEWORK-MIB SNMP-RAMEWORK-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB SNMP-USM-DRASD-SM-MIB	> Internet	Protoco	1 Version	4 Sec: 192 168 10	10 Det: 192 168 5 254			IPV6-MIB	
SIMP-RAAEWORK-MIB Simple Network Management Protocol Simple Network Sector Simple Network Sector Simpl	> liser Dat	agnam Pr	otocol S	rc Port: 62008 Dst	Port: 161			SNMP-COMMUNITY-MIB	
SINAP-AND SINAP-VIEW-BASED-SM-MIB SINAP-VIEW-BASED-SM-MIB SINAP-VIEW-BASED-SM-MIB CISCO-ENHANCED-MEMPOOL-MIB	> Simple I	latwork M	anagement	Protocol	FOTC. 101			SNMP-FRAMEWORK-MIB	
SNMP-NOXY-MIB SNMP-PROXY-MIB SNMP-VER-BASED-SM-MIB SNMP-USER-BASED-SM-MIB SNMP-USER-BASED-SM-MIB SNMP-VEW-BASED-ACM-MIB CISCO-ENHANCED-MEMPOOL-MIB	Junpaci	ICCHOIN II	onoBenetic	FICTOR				SNMP-MPD-MIB	
SNMP-PROXY-MIB SNMP-USER-BASED-SM-MIB SNMP-USER-BASED-SM-MIB SNMP-USER-BASED-ACM-MIB CISCO-ENHANCED-MEMPOOL-MIB								SNMP-NOTIFICATION-MIB	
SNMP-TARGET-MB SNMP-USM-DBECTS-MIB SNMP-USM-DH-OBJECTS-MIB SNMP-VIEW-BASED-ACM-MB CISCO-ENHANCED-MEMPOOL-MB CISCO-ENHANCED-MEMPOOL-MB CISCO-ENHANCED-MEMPOOL-MB OK Cancel Help								SNMP-PROXY-MIB	
SINIAP-USER-BASED-SM-IB SINIAP-USER-BASED-ACM-MIB CISCO-ENHANCED-MEMPOOL-MIB + - Pa V C C.1(deexs)apamox/AppOata_ing)Hitrenbackiam/_modules OK Cancel Help								SNMP-TARGET-MIB	
SNIMP-USW-BASED-ACM-MIB SNIMP-VIEW-BASED-ACM-MIB CISCO-ENHANCED-MEMPOOL-MIB * - % ~ Citteensigaamou'Age08ta_aig115treathariiom1_modules OK Cancel Help								SNMP-USER-BASED-SM-MIB	
SNMP-VIEW-BASED-ACM-MIB CISCO-ENHANCED-MEMPOOL-MIB + - B C Cilleersigaamoul4onDataingiltifreehankiemi modules OK Cancel Help								SNMP-USM-DH-OBJECTS-MIB	
CISCO-ENHANCED-MEMPOOL-MIB								SNMP-VIEW-BASED-ACM-MIB	
+ - R · · R C.iddersigaammuldoobetaingildirenharkismi modules OK Cancel Help								CISCO-ENHANCED-MEMPOOL-MIB	
+ - B								· · · · · · · · · · · · · · · · · · ·	
OK Cancel Help								+ - Pa A V C: Users igasimov AppDataing Wiresharkismi modules	
OK Cancel Help									
								OK Cancel Help	

3. Une fois Wireshark redémarré, la résolution OID est activée :



Sur la base de la sortie déchiffrée du fichier de capture, l'outil de surveillance SNMP interrogeait régulièrement (10 secondes d'intervalle) les données relatives à l'utilisation des pools de mémoire sur le FTD. Comme expliqué dans l'article TechNote <u>ASA SNMP Polling for Memory-Related</u> <u>Statistics</u>, l'interrogation de l'utilisation du pool partagé global (GSP) avec SNMP entraîne une utilisation CPU élevée. Dans ce cas, à partir des captures, il était clair que l'utilisation du pool partagé global était régulièrement interrogée dans le cadre de la primitive getBulkRequest SNMP.

Afin de minimiser les problèmes de CPU causés par le processus SNMP, il a été recommandé de suivre les étapes de mitigation pour les problèmes de CPU pour SNMP mentionnés dans l'article et d'éviter d'interroger les OID liés à GSP. Sans l'interrogation SNMP pour les OID qui se rapportent à GSP, aucun bogue de CPU causé par le processus SNMP n'a été observé et le taux de dépassements a diminué de manière significative.

Informations connexes

- Guides de configuration de Cisco Firepower Management Center
- Clarifier les actions de règle de politique de contrôle d'accès de Firepower Threat Defense
- Utiliser les captures Firepower Threat Defense et Packet Tracer
- <u>Découvrez Wireshark</u>

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.