Échec de la classification et de la détection des plug-ins P2P pour l'application avec les flux SSL dans ASR5x00

Contenu

Introduction

Ce document décrit un scénario spécifique dans lequel l'abonné utilise des applications gratuites telles que WhatsApp, Snapchat, etc. avec des flux SSL (Secure Sockets Layer) tout en bloquant le trafic d'autres utilisateurs. Cette application particulière s'exécute sur la gamme Cisco ASR 5x00. SSL est un protocole de réseau informatique qui gère l'authentification des serveurs, l'authentification des clients et la communication cryptée entre les serveurs et les clients.

Problème

Pour détecter une application, vous avez besoin de paquets initiaux pour l'analyse. Ces deux exigences contradictoires sont remplies dans toute la mesure possible.

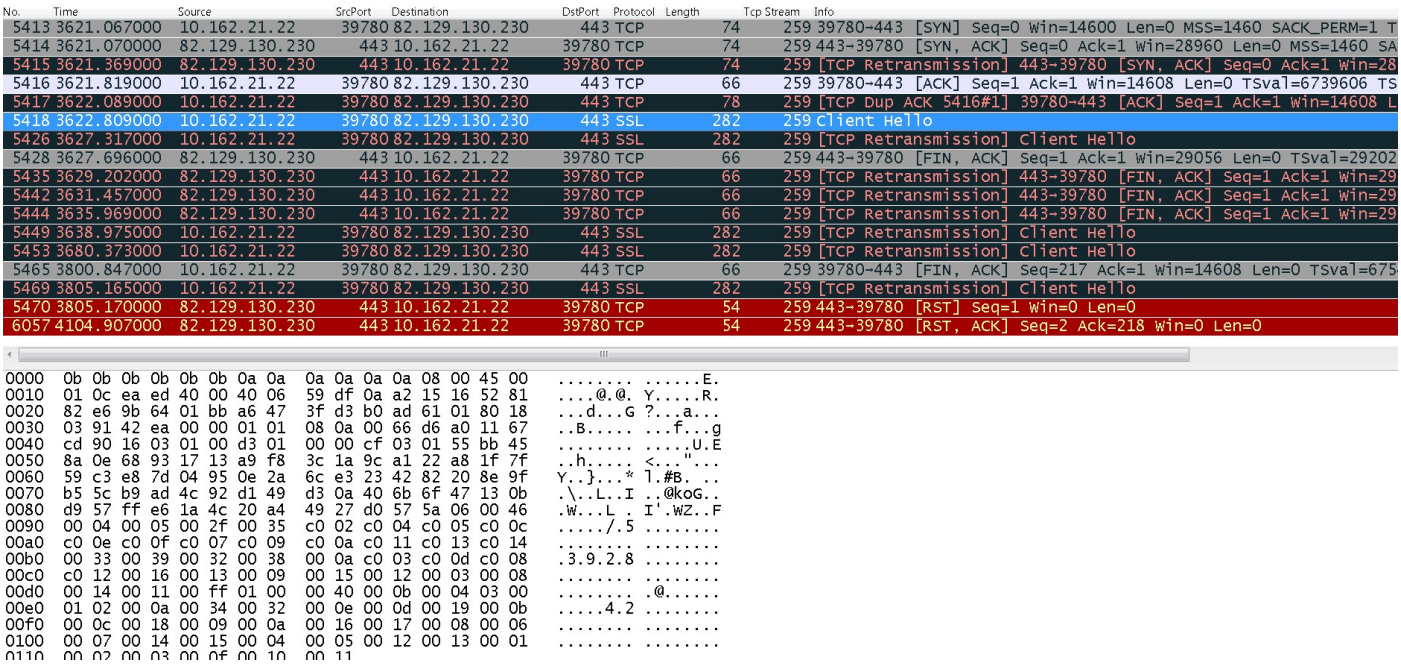
a) La détection doit se produire dans le premier paquet lui-même

b) La précision de détection doit être de 100 %

Si vous essayez de remplir la condition (a) et de marquer toutes les applications du premier paquet (ce qui n'est pas possible), la condition (b) sur la précision de détection souffre.Afin de rendre la précision de détection correcte, vous avez besoin de plus de paquets pour analyser beaucoup d'applications (il y a des applications et des flux où l'application est détectée dans le premier paquet lui-même). Dans le cas d'une même application, il peut arriver que vous puissiez marquer certains flux dans le premier paquet lui-même, alors que d'autres flux de la même application ont besoin de plus de paquets pour l'analyse.

Ainsi, si une application est gratuite tout en bloquant tout autre trafic, il peut arriver que le paquet initial de l'application ne soit pas détecté car il ne contient pas suffisamment d'informations. En particulier pour les applications basées sur les flux SSL, le protocole est marqué à l'aide du champ nom-serveur-indication présent dans le paquet hello-client ou du nom commun présent dans le certificat SSL. Comme le nom du serveur est un champ facultatif, il n'est pas toujours présent. Comme l'illustre cette image, dans un flux WhatsApp SSL, après une connexion en trois étapes (TWH), le paquet Hello du client est envoyé par l'application. **Une trace PCAP ne montrant aucun champ SNI (Server Name Indication). On peut également voir plusieurs retransmissions de**

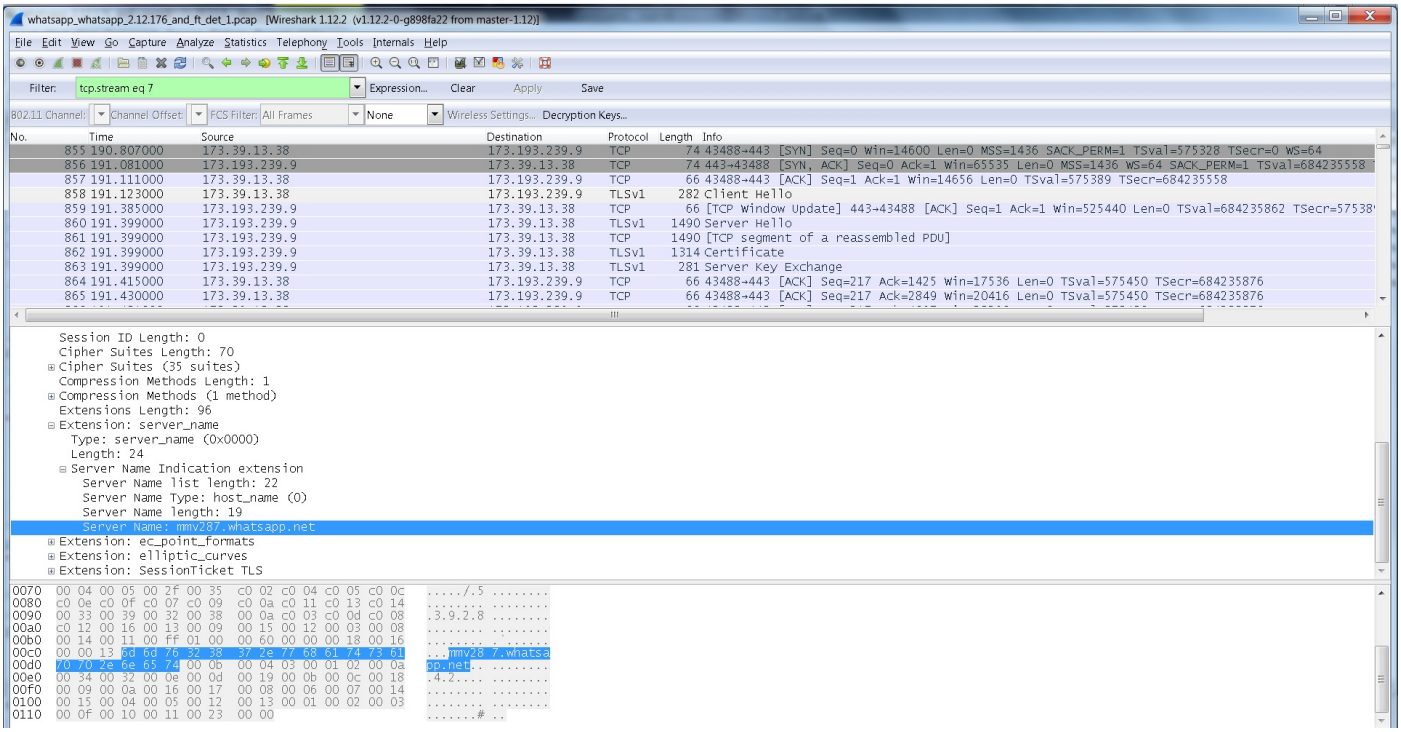
paquets Hello client qui finissent par être abandonnés.



No.	Time	Source	SrcPort	Destination	DstPort	Protocol	Length	Tcp Stream	Info
5413	3621.067000	10.162.21.22	39780	82.129.130.230	443	TCP	74	259	39780→443 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 T
5414	3621.070000	82.129.130.230	443	10.162.21.22	39780	TCP	74	259	443→39780 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA
5415	3621.369000	82.129.130.230	443	10.162.21.22	39780	TCP	74	259	[TCP Retransmission] 443→39780 [SYN, ACK] Seq=0 Ack=1 Win=28
5416	3621.819000	10.162.21.22	39780	82.129.130.230	443	TCP	66	259	39780→443 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=6739606 TS
5417	3622.089000	10.162.21.22	39780	82.129.130.230	443	TCP	78	259	[TCP Dup ACK 5416#1] 39780→443 [ACK] Seq=1 Ack=1 Win=14608 L
5418	3622.809000	10.162.21.22	39780	82.129.130.230	443	SSL	282	259	Client Hello
5426	3627.317000	10.162.21.22	39780	82.129.130.230	443	SSL	282	259	[TCP Retransmission] Client Hello
5428	3627.696000	82.129.130.230	443	10.162.21.22	39780	TCP	66	259	443→39780 [FIN, ACK] Seq=1 Ack=1 Win=29056 Len=0 TSval=29202
5435	3629.202000	82.129.130.230	443	10.162.21.22	39780	TCP	66	259	[TCP Retransmission] 443→39780 [FIN, ACK] Seq=1 Ack=1 Win=29
5442	3631.457000	82.129.130.230	443	10.162.21.22	39780	TCP	66	259	[TCP Retransmission] 443→39780 [FIN, ACK] Seq=1 Ack=1 Win=29
5444	3635.969000	82.129.130.230	443	10.162.21.22	39780	TCP	66	259	[TCP Retransmission] 443→39780 [FIN, ACK] Seq=1 Ack=1 Win=29
5449	3638.975000	10.162.21.22	39780	82.129.130.230	443	SSL	282	259	[TCP Retransmission] Client Hello
5453	3680.373000	10.162.21.22	39780	82.129.130.230	443	SSL	282	259	[TCP Retransmission] Client Hello
5465	3800.847000	10.162.21.22	39780	82.129.130.230	443	TCP	66	259	39780→443 [FIN, ACK] Seq=217 Ack=1 Win=14608 Len=0 TSval=675
5469	3805.165000	10.162.21.22	39780	82.129.130.230	443	SSL	282	259	[TCP Retransmission] Client Hello
5470	3805.170000	82.129.130.230	443	10.162.21.22	39780	TCP	54	259	443→39780 [RST] Seq=1 Win=0 Len=0
6057	4104.907000	82.129.130.230	443	10.162.21.22	39780	TCP	54	259	443→39780 [RST, ACK] Seq=2 Ack=218 Win=0 Len=0

```
0000  0b 0b 0b 0b 0b 0b 0a 0a  0a 0a 0a 0a 08 00 45 00   ........ ......E.
0010  01 0c ea ed 40 00 40 06  59 df 0a a2 15 16 52 81   ....@.@. Y.....R.
0020  82 e6 9b 64 01 bb a6 47  3f d3 b0 ad 61 01 80 18   ...d...G ?...a...
0030  03 91 42 ea 00 00 01 01  08 0a 00 66 d6 a0 11 67   ..B..... ...f...g
0040  cd 90 16 03 01 00 d3 01  00 00 cf 03 01 55 bb 45   ........ .....U.E
0050  8a 0e 68 93 17 13 a9 f8  3c 1a 9c a1 22 a8 1f 7f   ..h..... <..."...
0060  59 c3 e8 7d 04 95 0e 2a  6c e3 23 42 82 20 8e 9f   Y..}...* l.#B. ..
0070  b5 5c b9 ad 4c 92 d1 49  d3 0a 40 6b 6f 47 13 0b   .\..L..I ..@koG..
0080  d9 57 ff e6 1a 4c 20 a4  49 27 d0 57 5a 06 00 46   .W...L . I'.WZ..F
0090  00 04 00 05 00 2f 00 35  c0 02 c0 04 c0 05 c0 0c   ...../.5 ........
00a0  c0 0e c0 0f c0 07 c0 09  c0 0a c0 11 c0 13 c0 14   ........ ........
00b0  00 33 00 39 00 32 00 38  00 0a c0 03 c0 0d c0 08   .3.9.2.8 ........
00c0  c0 12 00 16 00 13 00 09  00 15 00 12 00 03 00 08   ........ ........
00d0  00 14 00 11 00 ff 01 00  00 40 00 0b 00 04 03 00   ........ .@......
00e0  01 02 00 0a 00 34 00 32  00 0e 00 0d 00 19 00 0b   .....4.2 ........
00f0  00 0c 00 18 00 09 00 0a  00 16 00 17 00 08 00 06   ........ ........
0100  00 07 00 14 00 15 00 04  00 05 00 12 00 13 00 01   ........ ........
0110  00 02 00 03 00 0f 00 10  00 11
```

En outre, comme l'illustre cette image, ils sont les hex-octets du paquet client-hello dans lequel le champ SNI, utilisé pour le marquage de WhatsApp, n'est pas présent. Par conséquent, le paquet client-hello ne peut pas être marqué comme WhatsApp et n'est pas détecté. Comme ce paquet entre dans un autre groupe de notation, il est abandonné et donc plusieurs retransmissions de paquet client-hello sont vues (voir trame n° 5449, 5453, 5469). Enfin, la connexion est interrompue. Plusieurs de ces flux sont observés dans le pcap. C'est la raison pour laquelle aucune activité utile, par exemple le téléchargement d'image pour WhatsApp, ne peut être effectuée.



Dépannage

```
1. capture  monitor subscriber imsi XXXX with following options
19 - User L3
X - PDU Hexdump
Verbosity level 5
```

Ces commandes donnent les statistiques de l'analyseur pour les applications.

```
# show act analyzer statistics name p2p application snapchat
# show act analyzer statistics name p2p application whatsapp
```

Pour vérifier la version du plug-in :

```
#show plugin p2p
Wednesday July 29 22:12:07 SAST 2015
plugin p2p
  patch-directory /var/opt/lib
  base-directory /lib
  base-version 1.50.52055
  module priority 1 version 1.139.505
```

Solution

Afin d'éviter, vous devez vous assurer que les paquets avant qu'une application (disons WhatApp) soit marquée et passe par.

Utilisez cette règle :

```
ruledef ssl_clienthello
      tcp either-port = 443
      tcp payload-length >= 44
      tcp payload  starts-with hex-signature 16-03
#exit
```

Tout paquet correspondant à la règle ci-dessus ne doit pas être abandonné. La priorité de cette règle doit être juste au-dessus de la règle par défaut (ip-any ruledef) qui correspondait à ce paquet et qui l'a fait tomber.

En utilisant cette configuration, seuls les paquets correspondant aux trois lignes de règle ci-dessus sont libres. Celles-ci incluent uniquement les paquets de connexion initiaux dans le flux SSL (tels que client-hello, server-hello) qui sont autorisés à utiliser cette règle, alors que tous les autres paquets dans le flux SSL ne correspondent pas à cette règle. Ainsi, s'il y a un SSLflow qui appartient à une autre application (autre que ce que vous voulez libérer), il ne peut pas y avoir de transaction utile, puisque seuls les deux à trois premiers paquets d'un flux SSL sont autorisés à utiliser cette règle.

Exemple de configuration

La règle suggérée doit avoir une priorité supérieure à all-ip_004_012_00016 ruledef (ip any-match = TRUE) et

action de charge qui autorise le trafic similaire à la règle whapp.(sid_040_rg_400_rate_99999/sid_040_rg_400_rate_00032/ sid_040_rg_400_rate_00 064 avec le groupe de notation 400 et tout tarif).

Avec cette configuration, le paquet Hello du client atteint la règle proposée et est autorisé plutôt

que redirigé. Voici les deux bases de règles où sont observées les règles de l'analyse des risques :

```
rulebase mbc-internet-rs action priority 1087 dynamic-only ruledef WhatsApp_P2P_040_400_99999_All_internet charging-action sid_040_rg_400_rate_99999 action priority 1088 dynamic-only ruledef WhatsApp_P2P_040_400_00064_All_internet charging-action sid_040_rg_400_rate_00064 action priority 1089 dynamic-only ruledef WhatsApp_P2P_040_400_00032_All_internet charging-action sid_040_rg_400_rate_00032 action priority [1090-9909] dynamic-only ruledef ssl_clienthello charging-action sid_040_rg_400_rate99999/00064/00032 --> Higher priority than all-ip ruledef and charging action with rating group 400
action priority 9910 dynamic-only ruledef all-ip_004_012_00016_MI_internet charging-action sid_004_rg_012_rate_00016
action priority 9920 dynamic-only ruledef all-ip_004_012_00032_MI_internet charging-action sid_004_rg_012_rate_00032
action priority 9930 dynamic-only ruledef all-ip_004_012_00064_MI_internet charging-action sid_004_rg_012_rate_00064


rulebase mbc-iphone-rs
action priority 1206 dynamic-only ruledef WhatsApp_P2P_040_400_99999_All_iphone charging-action sid_040_rg_400_rate_99999
action priority 1207 dynamic-only ruledef WhatsApp_P2P_040_400_00064_All_iphone charging-action sid_040_rg_400_rate_00064
action priority 1208 dynamic-only ruledef WhatsApp_P2P_040_400_00032_All_iphone charging-action sid_040_rg_400_rate_00032
action priority [1209-8999] dynamic-only ruledef ssl_clienthello charging-action sid_040_rg_400_rate99999/00064/00032 --> Higher priority than all-ip ruledef and charging action with rating group 400
action priority 9000 dynamic-only ruledef all-ip_015_150_00016_ALL_iphone charging-action sid_015_rg_150_rate_00016
action priority 9010 dynamic-only ruledef all-ip_015_150_00032_ALL_iphone charging-action sid_015_rg_150_rate_00032
action priority 9020 dynamic-only ruledef all-ip_015_150_00064_ALL_iphone charging-action sid_015_rg_150_rate_00064
action priority 9030 dynamic-only ruledef all-ip_015_150_99999_ALL_iphone charging-action sid_015_rg_150_rate_99999

charging-action sid_040_rg_400_rate_99999
content-id 400
service-identifier 40
billing-action egcdr
cca charging credit
exit

ruledef ssl_clienthello
tcp either-port = 443
tcp payload-length >= 44
tcp payload  starts-with hex-signature 16-03
exit
```