Contents

Introduction

This document describes a specific scenario in which the subscriber uses free-rate applications such as Whatsapp, Snapchat etc. with Secure Sockets Layer (SSL) flows while blocking other user traffic. This particular application runs on Cisco Aggregated Service Routers (ASR) 5x00 series. SSL is a computer networking protocol that manages server authentication, client authentication and encrypted communication between servers and clients.

Problem

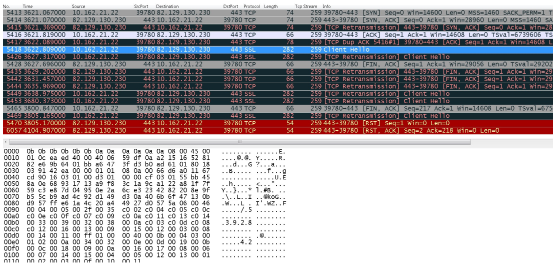
To detect any app, you need some initial packets for the analysis. These two contradictory requirements are fulfilled to the maximum extent possible.

a) Detection must happen in the first packet itself

b) Detection accuracy must be 100%

If you try to fullfill requirement (a) & mark all the apps in the first packet (that is not practically possible), the requirement (b) on detection accuracy suffers.In order to make the detection accuracy good, you need more packets to analyze lot of apps (there are apps & flows where the app is detected in the first packet itself). In case of the same app, it can happen that you are able to mark some flows in the first packet itself while other flows of the same app need more packets for analysis.

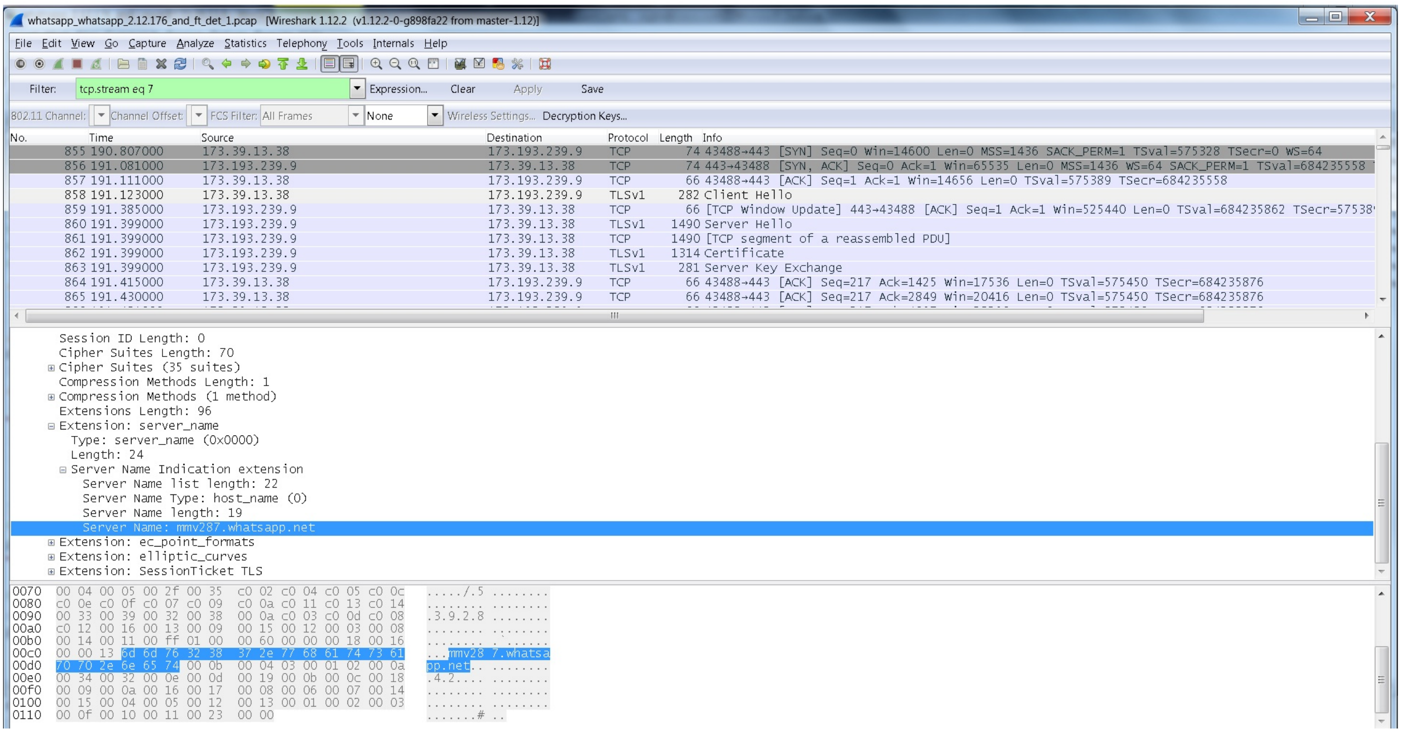
So if any of app is free-rated while blocking any other traffic, it can happen that the initial packet of the app does not get detected as it does not carry sufficient information. In particular case of apps based on SSL flows, the protocol is marked using either the server-name-indication field present in the client-hello packet or the common-name present in the SSL certificate. As the server-name is optional field, it is not always present. As shown in this image, in a Whatsapp SSL flow, after Three-Way-Handshake (TWH) the client hello packet is sent by the app. **A PCAP trace showing no Server Name Indication (SNI) field. Also seen are multiple retransmissions of client hello packets that eventually get dropped.**



```
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
```

 Also, as shown in this image, their are the hex-bytes for the client-hello packet in which the SNI field, used for marking Whatsapp, is not present. Hence, the client-hello packet cannot be marked as Whatsapp and goes undetected. As this packet falls into different rating-group, it gets dropped and hence multiple retransmissions of client-hello packet are seen (see frame no 5449, 5453, 5469). Finally, the connection gets terminated. Several such flows are seen in the pcap. This is the reason that no useful activity, for example the image upload for Whatsapp, can be
done.


whatsapp_whatsapp_2.12.176_and_ft_det_1.pcap [Wireshark 1.12.2 (v1.12.2-0-g898fa22 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames None Wireless Settings... Decryption Keys...

Filter: tcp.stream eq 7

```
No.     Time         Source           Destination      Protocol Length Info
    855 190.807000   173.39.13.38     173.193.239.9    TCP        74 43488→443 [SYN] Seq=0 Win=14600 Len=0 MSS=1436 SACK_PERM=1 TSval=575328 TSecr=0 WS=64
    856 191.081000   173.193.239.9    173.39.13.38     TCP        74 443→43488 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1436 WS=64 SACK_PERM=1 TSval=684235558
    857 191.111000   173.39.13.38     173.193.239.9    TCP        66 43488→443 [ACK] Seq=1 Ack=1 Win=14656 Len=0 TSval=575389 TSecr=684235558
    858 191.123000   173.39.13.38     173.193.239.9    TLSv1     282 Client Hello
    859 191.385000   173.193.239.9    173.39.13.38     TCP        66 [TCP Window Update] 443→43488 [ACK] Seq=1 Ack=1 Win=525440 Len=0 TSval=684235862 TSecr=57538
    860 191.399000   173.193.239.9    173.39.13.38     TLSv1    1490 Server Hello
    861 191.399000   173.193.239.9    173.39.13.38     TCP      1490 [TCP segment of a reassembled PDU]
    862 191.399000   173.193.239.9    173.39.13.38     TLSv1    1314 Certificate
    863 191.399000   173.193.239.9    173.39.13.38     TLSv1     281 Server Key Exchange
    864 191.415000   173.39.13.38     173.193.239.9    TCP        66 43488→443 [ACK] Seq=217 Ack=1425 Win=17536 Len=0 TSval=575450 TSecr=684235876
    865 191.430000   173.39.13.38     173.193.239.9    TCP        66 43488→443 [ACK] Seq=217 Ack=2849 Win=20416 Len=0 TSval=575450 TSecr=684235876
```

```
    Session ID Length: 0
    Cipher Suites Length: 70
  ⊞ Cipher Suites (35 suites)
    Compression Methods Length: 1
  ⊞ Compression Methods (1 method)
    Extensions Length: 96
  ⊟ Extension: server_name
      Type: server_name (0x0000)
      Length: 24
    ⊟ Server Name Indication extension
        Server Name list length: 22
        Server Name Type: host_name (0)
        Server Name length: 19
        Server Name: mmv287.whatsapp.net
  ⊞ Extension: ec_point_formats
  ⊞ Extension: elliptic_curves
  ⊞ Extension: SessionTicket TLS
```

```
0070  00 04 00 05 00 2f 00 35  c0 02 c0 04 c0 05 c0 0c   ...../.5 ........
0080  c0 0e c0 0f c0 07 c0 09  c0 0a c0 11 c0 13 c0 14   ........ ........
0090  00 33 00 39 00 32 00 38  00 0a c0 03 c0 0d c0 08   .3.9.2.8 ........
00a0  c0 12 00 16 00 13 00 09  00 15 00 12 00 03 00 08   ........ ........
00b0  00 14 00 11 00 ff 01 00  00 60 00 00 00 18 00 16   ........ ........
00c0  00 00 13 6d 6d 76 32 38  37 2e 77 68 61 74 73 61   ...mmv28 7.whatsa
00d0  70 70 2e 6e 65 74 00 0b  00 04 03 00 01 02 00 0a   pp.net.. ........
00e0  00 34 00 32 00 0e 00 0d  00 19 00 0b 00 0c 00 18   .4.2.... ........
00f0  00 09 00 0a 00 16 00 17  00 08 00 06 00 07 00 14   ........ ........
0100  00 15 00 04 00 05 00 12  00 13 00 01 00 02 00 03   ........ ........
0110  00 0f 00 10 00 11 00 23  00 00                     .......# ..
```

Troubleshoot

These commands give the analyser stats for the applications.

To check the plugin version:

Solution

In order to avoid, you need to ensure that the packets before an app (say whatsapp) get marked and must go through.

Use this ruledef :

Any packet, matching the above ruledef, must not be dropped. The priority of this ruledef must be just above the default ruledef (ip-any ruledef) that was matching this packet & causing it to get dropped.

By using this configuration, only the packets matching the above three rule-lines are free-rated. These include only the initial handshake packets in SSL flow (such as client-hello, server-hello) that are allowed using this ruledef, while all other packets in SSL flow do not match this ruledef. Thus, if there is a SSLflow that belongs to some other app (other than whatsapp that you want to free-rate), there cannot be any useful transaction, since only the initial two to three packets of an SSL flow are allowed to use this ruledef.

Sample Configuration

The suggested ruledef needs to have a higher priority than all-ip_004_012_00016 ruledef (ip any-match = TRUE) and

charging action that allows the traffic similar to whatsapp ruledef.(sid_040_rg_400_rate_99999/sid_040_rg_400_rate_00032/ sid_040_rg_400_rate_00064 with rating-group 400 and any rate).

With this config, the client hello packet hits the proposed ruledef and is allowed rather than being redirected. These are the two rulebases where whatsapp rules are seen:

```
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
```