Classificazione e rilevamento plug-in P2P non riusciti per l'applicazione con flussi SSL in ASR5x00

Sommario

[Introduzione](#)
[Problema](#)
[Risoluzione dei problemi](#)
[Soluzione](#)
[Esempio di configurazione](#)
[Discussioni correlate nella Cisco Support Community](#)

Introduzione

Questo documento descrive uno scenario specifico in cui il sottoscrittore usa applicazioni a velocità libera come Whatsapp, Snapchat ecc. con flussi SSL (Secure Sockets Layer) bloccando il traffico di altri utenti. Questa particolare applicazione viene eseguita su Cisco Aggregated Service Router (ASR) serie 5x00. SSL è un protocollo di rete per computer che gestisce l'autenticazione server, l'autenticazione client e la comunicazione crittografata tra server e client.

Problema

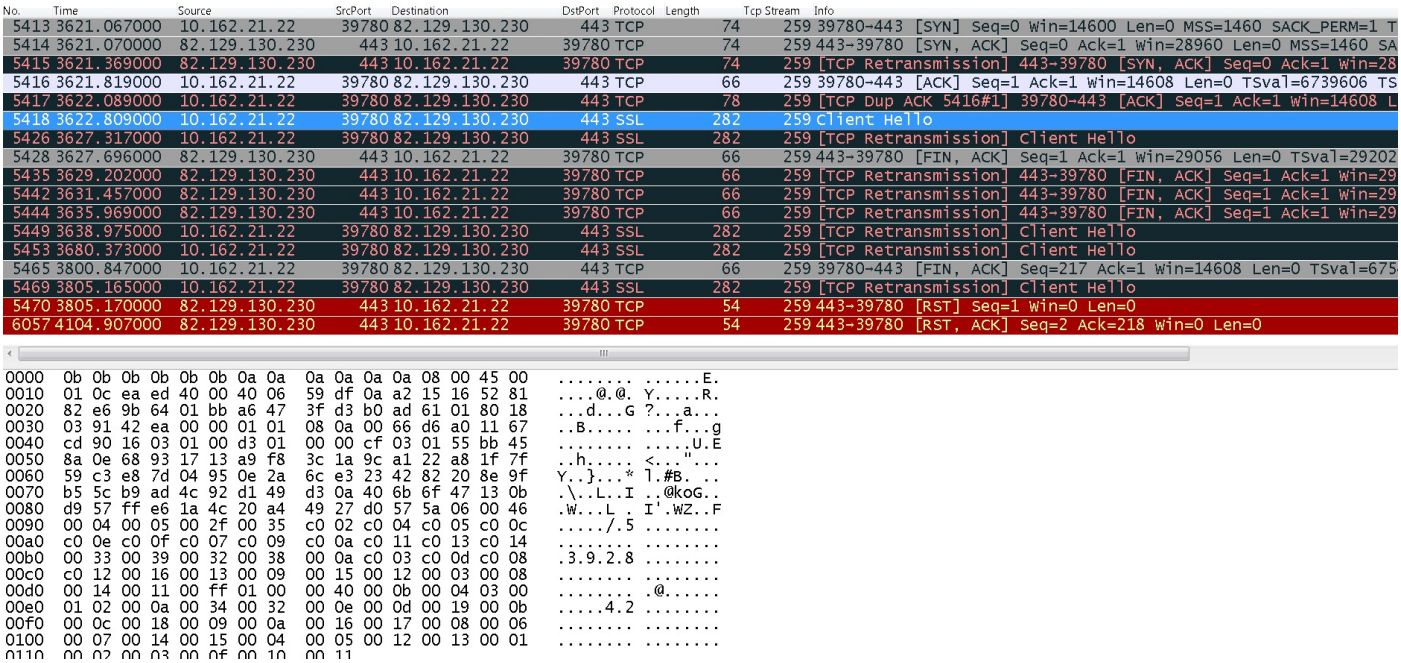
Per rilevare qualsiasi app, sono necessari alcuni pacchetti iniziali per l'analisi. Queste due esigenze contraddittorie sono soddisfatte nella misura massima possibile.

a) Il rilevamento deve avvenire nel primo pacchetto

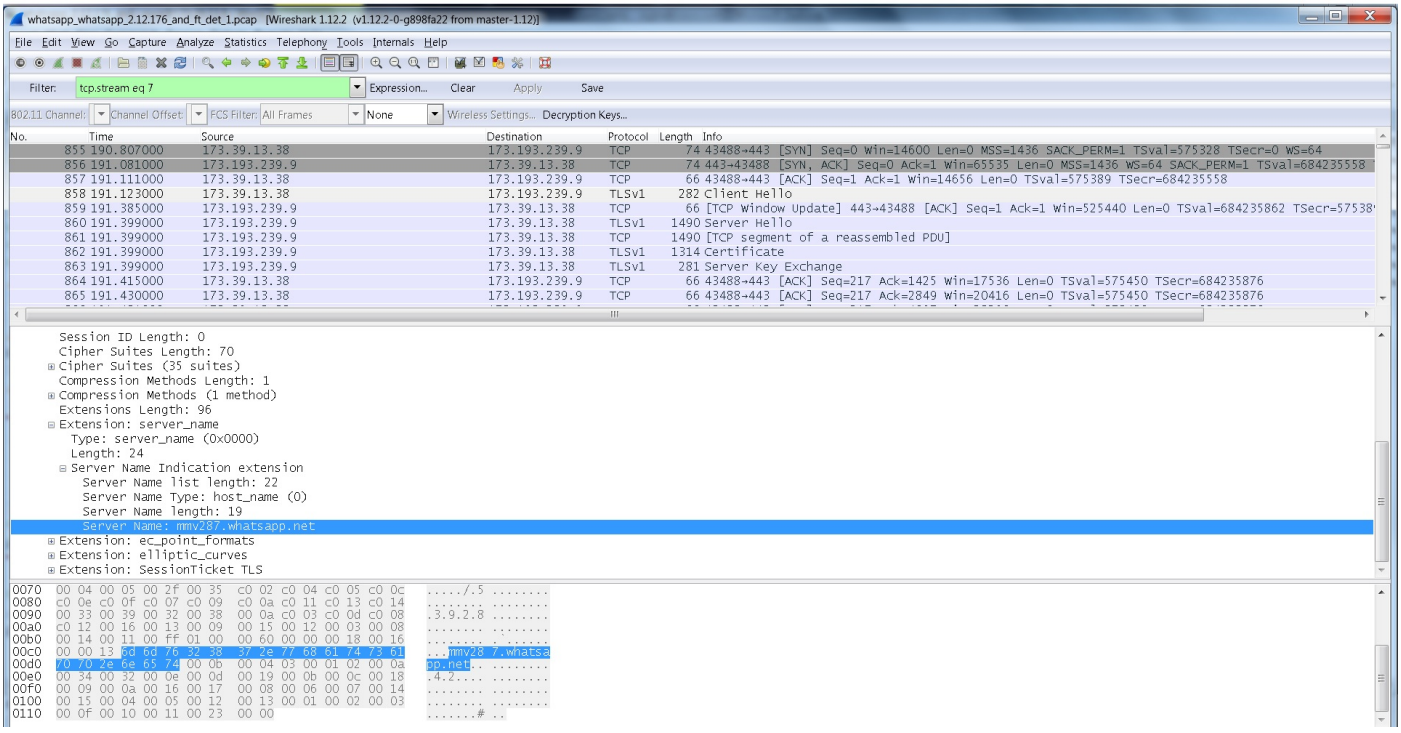
b) La precisione del rilevamento deve essere del 100%

Se si tenta di soddisfare il requisito (a) e contrassegnare tutte le applicazioni nel primo pacchetto (che non è praticamente possibile), il requisito (b) sulla precisione del rilevamento subisce.Per rendere ottimale la precisione del rilevamento, sono necessari più pacchetti per analizzare molte applicazioni (ci sono app e flussi in cui l'applicazione viene rilevata nel primo pacchetto stesso). Nel caso della stessa app, può accadere che si sia in grado di contrassegnare alcuni flussi nel primo pacchetto stesso, mentre altri flussi della stessa app richiedono più pacchetti per l'analisi.

Quindi se una qualsiasi delle app è gratuita mentre blocca qualsiasi altro traffico, può succedere che il pacchetto iniziale dell'app non venga rilevato in quanto non contiene informazioni sufficienti. In particolare, nel caso di applicazioni basate su flussi SSL, il protocollo è contrassegnato utilizzando il campo di indicazione del nome del server presente nel pacchetto client-hello o il nome comune presente nel certificato SSL. Poiché il campo server-name è facoltativo, non è sempre presente. Come mostrato in questa immagine, in un flusso SSL Whatsapp, dopo Three-Way-Handshake (TWH) il pacchetto hello del client viene inviato dall'app. **Traccia PCAP priva di campo SNI (Server Name Indication). Sono inoltre presenti più ritrasmissioni di pacchetti hello del client che alla fine vengono scartati.**



Inoltre, come mostrato in questa immagine, sono i byte esadecimali del pacchetto client-hello in cui il campo SNI, usato per contrassegnare Whatsapp, non è presente. Pertanto, il pacchetto client-hello non può essere contrassegnato come Whatsapp e non viene rilevato. Poiché il pacchetto rientra in un gruppo di classificazione diverso, viene scartato e si verificano quindi più ritrasmissioni di un pacchetto client-hello (vedere i frame n. 5449, 5453, 5469). Infine, la connessione viene terminata. Diversi flussi di questo tipo sono visibili nella capsula. Per questo motivo non è possibile eseguire alcuna attività utile, ad esempio il caricamento di immagini per Whatsapp.



Risoluzione dei problemi

1. capture monitor subscriber imsi XXXX with following options
19 - User L3

```
X - PDU Hexdump
Verbosity level 5
```
Questi comandi forniscono lo stato dell'analizzatore per le applicazioni.

```
# show act analyzer statistics name p2p application snapchat
# show act analyzer statistics name p2p application whatsapp
```
Per controllare la versione del 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
```

Soluzione

Per evitarlo, devi assicurarti che i pacchetti prima che un'app (ad esempio whatsapp) venga contrassegnata e debba passare.

Utilizza il seguente oggetto ruledef:

```
ruledef ssl_clienthello
      tcp either-port = 443
      tcp payload-length >= 44
      tcp payload  starts-with hex-signature 16-03
#exit
```
I pacchetti che corrispondono alla definizione di regola precedente non devono essere eliminati. La priorità di questo oggetto ruledef deve essere immediatamente superiore all'oggetto ruledef predefinito (ip-any ruledef) corrispondente al pacchetto e causarne l'eliminazione.

Utilizzando questa configurazione, solo i pacchetti che corrispondono alle tre righe sopra menzionate sono a tariffa libera. Tra questi sono inclusi solo i pacchetti di handshake iniziali nel flusso SSL (ad esempio client-hello, server-hello) consentiti tramite questo oggetto ruledef, mentre tutti gli altri pacchetti nel flusso SSL non corrispondono a questo oggetto ruledef. Pertanto, se esiste un SSLflow che appartiene a qualche altra app (diversa da whatsapp che si desidera rendere disponibile), non può esserci alcuna transazione utile, poiché solo i due o tre pacchetti iniziali di un SSL flow possono utilizzare questo ruledef.

Esempio di configurazione

Il ruledef suggerito deve avere una priorità più alta di all-ip_004_012_00016 ruledef (ip any-match = TRUE) e

operazione di caricamento che consente un traffico simile al traffico di whatsapp ruledef.(sid_040_rg_400_rate_9999/sid_040_rg_400_rate_00032/ sid_040_rg_400_rate_00064 con gruppo di classificazione 400 e qualsiasi tariffa).

Con questa configurazione, il pacchetto hello del client raggiunge il ruledef proposto ed è consentito anziché essere reindirizzato. Queste sono le due basi delle regole in cui vengono visualizzate le regole whatsapp:

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