

Dépannage de la gestion des chemins de données par UTD et filtrage des URL

Contenu

[Introduction](#)

[Informations générales](#)

[Vue de haut niveau Datapath](#)

[Du LAN/WAN au conteneur](#)

[Du conteneur au LAN/WAN](#)

[Plongée en profondeur Datapath](#)

[Paquet entrant du côté LAN ou WAN vers le conteneur](#)

[Paquet entrant du conteneur vers le côté LAN ou WAN](#)

[Intégration de la journalisation de flux UTD avec Packet-trace](#)

[Requête préalable :](#)

[Vérification de la compatibilité de la version UTD avec IOS XE](#)

[Vérifier la configuration du serveur de noms valide dans le conteneur](#)

[Problème 1](#)

[Dépannage](#)

[Cause première](#)

[Problème 2](#)

[Dépannage](#)

[Cause première](#)

[Problème 3](#)

[Dépannage](#)

[Étape 1 : Collecte de statistiques générales](#)

[Étape2: Affichage du fichier journal des applications](#)

[Problème 4](#)

[Dépannage](#)

[Cause première](#)

[Références](#)

Introduction

Ce document décrit comment dépanner Unified Threat Defense (UTD) également appelé Snort et Uniform Resource Locator (URL) Filtering sur les routeurs WAN Edge IOS[®] XE.

Informations générales

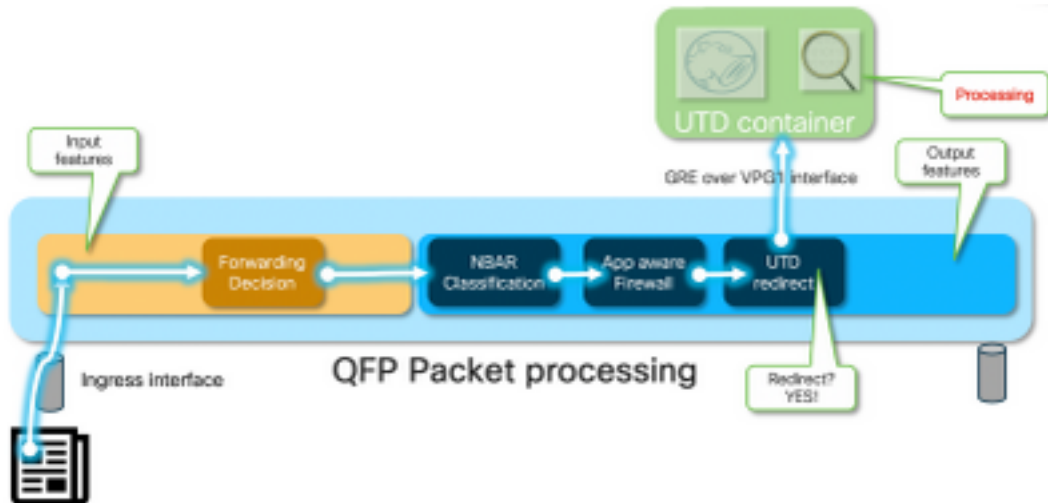
Snort est le système de prévention des intrusions (IPS) le plus répandu au monde. Depuis 2013, la société qui a créé une version commerciale du logiciel Snort, Sourcefire, est acquise par Cisco. À partir de la version 16.10.1 du logiciel SD-WAN IOS[®] XE, des conteneurs UTD/URF-Filtering ont été ajoutés à la solution Cisco SD-WAN.

Le conteneur s'enregistre sur le routeur IOS® XE à l'aide du framework app-nav. L'explication de ce processus dépasse le cadre du présent document.

Vue de haut niveau Datapath

À un niveau élevé, le chemin de données ressemble à ceci :

Du LAN/WAN au conteneur



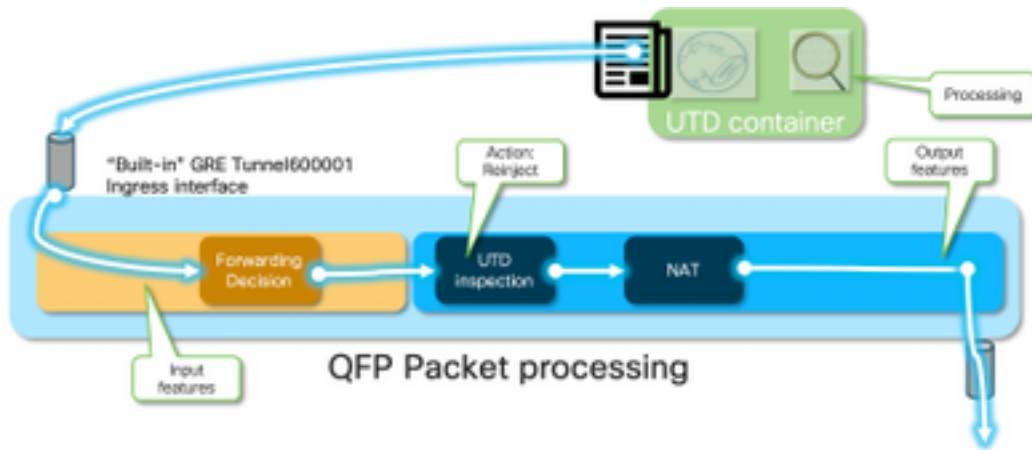
Le trafic provient du côté LAN. Comme IOS® XE sait que le conteneur est en bonne santé, il dévie le trafic vers le conteneur UTD. La dérivation utilise l'interface VirtualPortGroup1 comme interface de sortie, qui encapsule le paquet à l'intérieur d'un tunnel GRE (Generic Routing Encapsulation).

Le routeur exécute l'action PUNT en utilisant la cause :64 (paquet du moteur de service) et envoie le trafic vers le processeur de routage (RP). Un en-tête de point est ajouté et le paquet est envoyé au conteneur à l'aide d'une interface de sortie interne vers le conteneur "[internal0/0/svc_eng:0]"

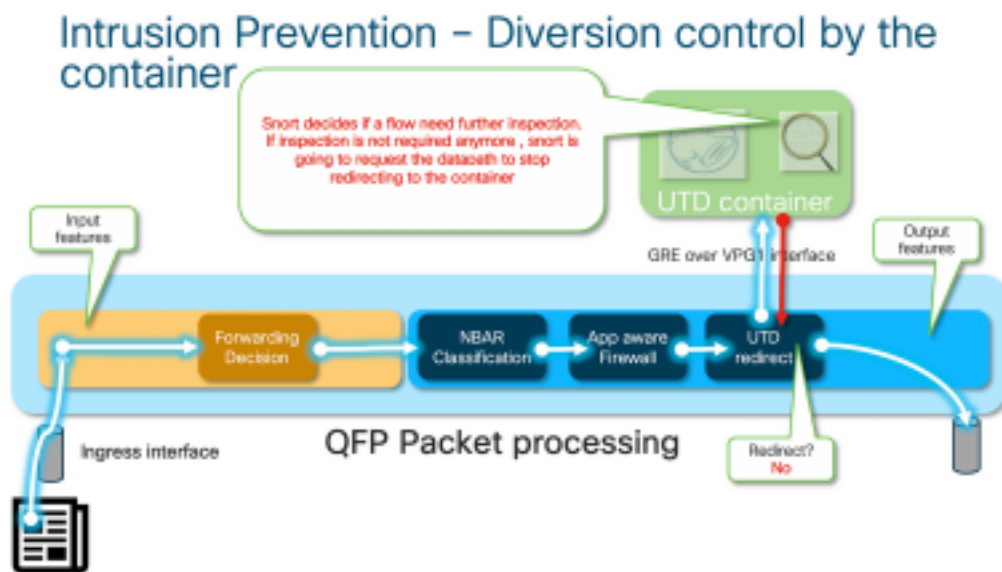
À ce stade, Snort tire parti de ses préprocesseurs et de ses ensembles de règles. Le paquet peut être abandonné ou transféré en fonction des résultats du traitement.

Du conteneur au LAN/WAN

En supposant que le trafic n'est pas censé être abandonné, le paquet est renvoyé au routeur après le traitement UTD. Il apparaît sur le processeur de flux quantique (QFP) comme provenant du tunnel6000001. Ensuite, il est traité par le routeur et doit être (espérons-le) routé vers l'interface WAN.



Le conteneur contrôle le résultat de la dérivation dans l'inspection UTD du chemin de données IOS® XE.



Par exemple, avec le flux HTTPS, les préprocesseurs sont intéressés de voir les paquets Hello du serveur / Hello du client avec négociation TLS. Par la suite, le flux n'est pas redirigé, car l'inspection du trafic chiffré TLS présente peu de valeur.

Plongée en profondeur Datapath

Du point de vue de packet-tracer, cet ensemble d'actions va être visible (192.168.16.254 est un client Web) :

```
debug platform condition ipv4 192.168.16.254/32 both
debug platform condition start
debug platform packet-trace packet 256 fia-trace data-size 3000
```

Paquet entrant du côté LAN ou WAN vers le conteneur

Dans ce scénario particulier, le paquet suivi provient du réseau local. Du point de vue de la redirection, il existe des différences pertinentes si le flux provient d'un réseau local ou étendu.

Le client tente d'accéder à www.cisco.com sur HTTPS

```

cedge6#show platform packet-trace packet 14
Packet: 14          CBUG ID: 3849209
Summary
  Input      : GigabitEthernet2
  Output     : internal0/0/svc_eng:0
  State      : PUNT 64 (Service Engine packet)
  Timestamp
    Start    : 1196238208743284 ns (05/08/2019 10:50:36.836575 UTC)
    Stop     : 1196238208842625 ns (05/08/2019 10:50:36.836675 UTC)
Path Trace
  Feature: IPV4(Input)
    Input      : GigabitEthernet2
    Output     : <unknown>
    Source     : 192.168.16.254
    Destination : 203.0.113.67
    Protocol   : 6 (TCP)
    SrcPort    : 35568
    DstPort    : 443
  Feature: DEBUG_COND_INPUT_PKT
    Entry      : Input - 0x8177c67c
    Input      : GigabitEthernet2
    Output     : <unknown>
    Lapsed time : 2933 ns

```

<snip>

Le trafic correspondant à la condition est suivi en entrée sur l'interface GigabitEthernet2.

```

Feature: UTD Policy (First FIA)
  Action      : Divert
  Input interface : GigabitEthernet2
  Egress interface: GigabitEthernet3
Feature: OUTPUT_UTD_FIRST_INSPECT
  Entry      : Output - 0x817cc5b8
  Input      : GigabitEthernet2
  Output     : GigabitEthernet3
  Lapsed time : 136260 ns
Feature: UTD Inspection
  Action      : Divert          <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  Input interface : GigabitEthernet2
  Egress interface: GigabitEthernet3
Feature: OUTPUT_UTD_FINAL_INSPECT
  Entry      : Output - 0x817cc5e8
  Input      : GigabitEthernet2
  Output     : GigabitEthernet3
  Lapsed time : 43546 ns

```

<snip>

Sur la baie d'appel des fonctionnalités de sortie (FIA) de l'interface de sortie, UTD FIA a décidé de détourner ce paquet vers le conteneur.

```

Feature: IPV4_OUTPUT_LOOKUP_PROCESS_EXT
  Entry      : Output - 0x81781bb4
  Input      : GigabitEthernet2
  Output     : Tunnel6000001
<removed>
Feature: IPV4_OUTPUT_LOOKUP_PROCESS_EXT
  Entry      : Output - 0x81781bb4
  Input      : GigabitEthernet2
  Output     : Tunnel6000001
<removed>
Feature: IPV4_INPUT_LOOKUP_PROCESS_EXT

```

```
Entry      : Output - 0x8177c698
Input      : Tunnel6000001
Output     : VirtualPortGroup1
Lapsed time : 880 ns
<snip>
```

Le paquet est placé sur le tunnel par défaut Tunnel600001 et est routé via l'interface VPG1. À ce stade, le paquet d'origine est encapsulé GRE.

```
Feature: OUTPUT_SERVICE_ENGINE
Entry      : Output - 0x817c6b10
Input      : Tunnel6000001
Output     : internal0/0/svc_eng:0
Lapsed time : 15086 ns
<removed>
```

```
Feature: INTERNAL_TRANSMIT_PKT_EXT
Entry      : Output - 0x8177c718
Input      : Tunnel6000001
Output     : internal0/0/svc_eng:0
Lapsed time : 43986 ns
```

Le paquet est transmis en interne au conteneur.

Note: Des renseignements supplémentaires sur les contenants internes sont fournis à titre d'information seulement. Le conteneur UTD n'est pas accessible via l'interface de ligne de commande normale.

Plus profondément dans le routeur lui-même, le trafic arrive dans un VRF interne sur l'interface eth2 du processeur de routage :

```
[cedge6:/$] chvrf utd ifconfig
eth0      Link encap:Ethernet HWaddr 54:0e:00:0b:0c:02
          inet6 addr: fe80::560e:ff:fe0b:c02/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1375101 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1366614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:96520127 (92.0 MiB) TX bytes:96510792 (92.0 MiB)

eth1      Link encap:Ethernet HWaddr 00:1e:e6:61:6d:ba
          inet addr:192.168.1.2 Bcast:192.168.1.3 Mask:255.255.255.252
          inet6 addr: fe80::21e:e6ff:fe61:6dba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:2000 Metric:1
          RX packets:1069 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2001 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:235093 (229.5 KiB) TX bytes:193413 (188.8 KiB)

eth2      Link encap:Ethernet HWaddr 00:1e:e6:61:6d:b9
          inet addr:192.0.2.2 Bcast:192.0.2.3 Mask:255.255.255.252
          inet6 addr: fe80::21e:e6ff:fe61:6db9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:2000 Metric:1
          RX packets:2564233 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2564203 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:210051658 (200.3 MiB) TX bytes:301467970 (287.5 MiB)

lo        Link encap:Local Loopback
```

```

inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

```

Eth0 est une interface TIPC (Transport Inter Process Communication) connectée au processus IOSd. Le canal OneP s'exécute sur lui pour transmettre des configurations et des notifications entre IOSd et le conteneur UTD.

De ce qui vous préoccupe, « eth2 [interface de conteneur]" est ponté à « VPG1 [192.0.2.1/192.168.2.2]" sont les adresses poussées par vManage vers l'IOS-XE et le conteneur.

Si vous exécutez **tcpdump**, vous pouvez voir le trafic encapsulé GRE allant au conteneur. L'encapsulation GRE inclut un en-tête VPATH.

```

[cedge6:/]$ chvrf utd tcpdump -nNvvvXi eth2 not udp
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
06:46:56.350725 IP (tos 0x0, ttl 255, id 35903, offset 0, flags [none], proto GRE (47), length 121)
  192.0.2.1 > 192.0.2.2: GREv0, Flags [none], length 101
gre-proto-0x8921
0x0000:  4500 0079 8c3f 0000 ff2f ab12 c000 0201  E..y.?.../.....
0x0010:  c000 0202 0000 8921 4089 2102 0000 0000  .....!@!.....
0x0020:  0000 0000 0300 0001 0000 0000 0000 0000  .....
0x0030:  0004 0800 e103 0004 0008 0000 0001 0000  .....
0x0040:  4500 0039 2542 4000 4011 ce40 c0a8 10fe  E..9%B@.@@....
0x0050:  ad26 c864 8781 0035 0025 fe81 cfa8 0100  .&.d...5%.
0x0060:  0001 0000 0000 0000 0377 7777 0363 6e6e  .....www.cnn
0x0070:  0363 6f6d 0000 0100 01                                .com.....

```

Paquet entrant du conteneur vers le côté LAN ou WAN

Après le traitement Snort (en supposant que le trafic ne doit pas être abandonné), il est réinjecté dans le chemin de transfert QFP.

```

cedge6#show platform packet-trace packet 15
Packet: 15          CBUG ID: 3849210
Summary
  Input       : Tunnel6000001
  Output      : GigabitEthernet3
  State       : FWD

```

Tunnel600001 est l'interface de sortie du conteneur.

```

Feature: OUTPUT_UTD_FIRST_INSPECT_EXT
  Entry       : Output - 0x817cc5b8
  Input       : GigabitEthernet2
  Output      : GigabitEthernet3
  Lapsed time : 2680 ns
Feature: UTD Inspection
  Action      : Reinject
  Input interface : GigabitEthernet2
  Egress interface: GigabitEthernet3
Feature: OUTPUT_UTD_FINAL_INSPECT_EXT

```

```
Entry      : Output - 0x817cc5e8
Input      : GigabitEthernet2
Output     : GigabitEthernet3
Lapsed time : 12933 ns
```

Comme le trafic a déjà été inspecté, le routeur sait qu'il s'agit d'une réinjection.

```
Feature: NAT
Direction : IN to OUT
Action     : Translate Source
Steps     :
Match id   : 1
Old Address : 192.168.16.254 35568
New Address : 172.16.16.254 05062
```

Le trafic reçoit NATed et va vers Internet.

```
Feature: MARMOT_SPA_D_TRANSMIT_PKT
Entry      : Output - 0x8177c838
Input      : GigabitEthernet2
Output     : GigabitEthernet3
Lapsed time : 91733 ns
```

Intégration de la journalisation de flux UTD avec Packet-trace

IOS-XE 17.5.1 a ajouté l'intégration de la journalisation de flux UTD avec packet-trace, où la sortie path-trace inclura un verdict UTD. Le verdict peut être l'un des suivants, par exemple :

- Le paquet que UTD décide de bloquer/d'alerter pour Snort
- autoriser/abandonner pour URLF
- bloquer/autoriser AMP

Pour les paquets qui n'ont pas les informations de verdict UTD, aucune information de journalisation de flux n'est consignée. Notez également qu'il n'existe aucune journalisation du verdict d'autorisation/d'autorisation IPS/IDS en raison d'un impact négatif potentiel sur les performances.

Pour activer l'intégration de la journalisation des flux, utilisez le modèle de module complémentaire CLI avec :

```
utd engine standard multi-tenancy
utd global
  flow-logging all
```

Exemple de sortie pour différents verdicts :

Délai de recherche d'URL :

```
show platform packet-trace pack all | sec Packet: | Feature: UTD Inspection
Packet: 31          CBUG ID: 12640
Feature: UTD Inspection
  Action              : Reinject
  Input interface     : GigabitEthernet2
  Egress interface    : GigabitEthernet3
  Flow-Logging Information :
  URLF Policy ID      : 1
```

```
URLF Action          : Allow(1)
URLF Reason           : URL Lookup Timeout(8)
```

Réputation et verdict URLF Autoriser :

```
Packet: 21          CBUG ID: 13859
Feature: UTD Inspection
Action              : Reinject
Input interface     : GigabitEthernet3
Egress interface    : GigabitEthernet2
Flow-Logging Information :
URLF Policy ID      : 1
URLF Action         : Allow(1)
URLF Reason         : No Policy Match(4)
URLF Category       : News and Media(63)
URLF Reputation     : 81
```

Réputation et verdict URLF Bloquer :

```
Packet: 26          CBUG ID: 15107
Feature: UTD Inspection
Action              : Reinject
Input interface     : GigabitEthernet3
Egress interface    : GigabitEthernet2
Flow-Logging Information :
URLF Policy ID      : 1
URLF Action         : Block(2)
URLF Reason         : Category/Reputation(3)
URLF Category       : Social Network(14)
URLF Reputation     : 81
```

Requête préalable :

Vérification de la compatibilité de la version UTD avec IOS XE

```
cedge7#sh utd eng sta ver
UTD Virtual-service Name: utd
IOS-XE Recommended UTD Version: 1.10.33_SV2.9.16.1_XEmain
IOS-XE Supported UTD Regex: ^1\.10\.([0-9]+)_SV(.*?)_XEmain$
UTD Installed Version: 1.0.2_SV2.9.16.1_XE17.5 (UNSUPPORTED)
```

Si l'option UNSUPPORTED (NON SUPPORTÉ) s'affiche, la mise à niveau du conteneur est requise en première étape avant de commencer le dépannage.

Vérifier la configuration du serveur de noms valide dans le conteneur

Certains services de sécurité tels qu'AMP et URLF nécessitent que le conteneur UTD soit capable de résoudre les noms des fournisseurs de services cloud. Par conséquent, le conteneur UTD doit avoir des configurations de serveur de noms valides. Pour vérifier cela, il suffit de vérifier le fichier `solvv.conf` du conteneur sous l'interpréteur de commandes système :

```
cedge:/harddisk/virtual-instance/utd/rootfs/etc]$ more resolv.conf
nameserver 208.67.222.222
nameserver 208.67.220.220
nameserver 8.8.8.8
```


Problème 1

Par conception, Unified Thread Defense doit être entièrement configuré avec le cas d'utilisation de Direct Internet Access (DIA). Le conteneur tentera de résoudre **api.bcti.brightcloud.com** afin d'interroger les réputations et catégories d'URL. Dans cet exemple, aucune des URL inspectées n'est bloquée même si la configuration appropriée est appliquée

Dépannage

Regardez toujours le fichier journal du conteneur.

```
cedge6#app-hosting move appid utd log to bootflash:  
Successfully moved tracelog to bootflash:  
iox_utd_R0-0_R0-0.18629_0.20190501005829.bin.gz
```

qui copie le fichier journal sur la mémoire Flash elle-même.

L'affichage du journal peut être réalisé à l'aide de la commande suivante :

```
cedge6# more /compressed iox_utd_R0-0_R0-0.18629_0.20190501005829.bin.gz
```

L'affichage du journal révèle :

```
2019-04-29 16:12:12 ERROR: Cannot resolve host api.bcti.brightcloud.com: Temporary failure in  
name resolution  
2019-04-29 16:17:52 ERROR: Cannot resolve host api.bcti.brightcloud.com: Temporary failure in  
name resolution  
2019-04-29 16:23:32 ERROR: Cannot resolve host api.bcti.brightcloud.com: Temporary failure in  
name resolution  
2019-04-29 16:29:12 ERROR: Cannot resolve host api.bcti.brightcloud.com: Temporary failure in  
name resolution  
2019-04-29 16:34:52 ERROR: Cannot resolve host api.bcti.brightcloud.com: Temporary failure in  
name resolution  
2019-04-29 16:40:27 ERROR: Cannot resolve host api.bcti.brightcloud.com: Temporary failure in  
name resolution
```

Par défaut, vManage provisionne un conteneur qui utilise le serveur OpenDNS [208.67.222.222 et 208.67.220.220]

Cause première

Le trafic DNS (Domain Name System) pour résoudre **api.bcti.brightcloud.com** est abandonné quelque part dans le chemin entre le conteneur et les serveurs DNS parapluie. Assurez-vous toujours que les deux DNS sont accessibles.

Problème 2

Dans un scénario où les sites Web de la catégorie Informations sur l'ordinateur et Internet sont censés être bloqués, la demande http à www.cisco.com est correctement abandonnée alors que les demandes HTTPS ne le sont pas.

Dépannage

[CSCvo77664](#) « Le filtrage URL UTD pour la recherche de catégorie échoue avec l'échec de recherche webroot » est à propos de la fuite du trafic lorsque le logiciel n'a pas encore de réponse à notre demande de verdict d'URL.

Problème 3

Dans ce scénario, par intermittence, les sessions de navigation Web qui devraient être autorisées par le filtrage d'URL [en raison de leur classification] sont supprimées. Par exemple, l'accès à [www.google.com](#) est aléatoire même si la catégorie « moteur de recherche web » est autorisée.

Dépannage

Étape 1 : Collecte de statistiques générales

Remarque Cette sortie de commande est réinitialisée toutes les 5 minutes

```
cedge7#show utd engine standard statistics internal
*****Engine #1*****
<removed> ===== HTTP
Inspect - encodings (Note: stream-reassembled packets included): <<<<<<<<< generic layer7 HTTP
statistics POST methods: 0 GET methods: 7 HTTP Request Headers extracted: 7 HTTP Request Cookies
extracted: 0 Post parameters extracted: 0 HTTP response Headers extracted: 6 HTTP Response
Cookies extracted: 0 Unicode: 0 Double unicode: 0 Non-ASCII representable: 0 Directory
traversals: 0 Extra slashes ("/"): 0 Self-referencing paths ("."): 0 HTTP Response Gzip
packets extracted: 0 Gzip Compressed Data Processed: n/a Gzip Decompressed Data Processed: n/a
Http/2 Rebuilt Packets: 0 Total packets processed: 13 <removed>
===== SSL
Preprocessor: <<<<<<<<< generic layer7 SSL statistics SSL packets decoded: 38 Client Hello: 8
Server Hello: 8 Certificate: 2 Server Done: 6 Client Key Exchange: 2 Server Key Exchange: 2
Change Cipher: 10 Finished: 0 Client Application: 2 Server Application: 11 Alert: 0 Unrecognized
records: 11 Completed handshakes: 0 Bad handshakes: 0 Sessions ignored: 4 Detection disabled: 1

<removed> UTM Preprocessor Statistics < URL filtering statistics including -----
----- URL Filter Requests Sent: 11 URL Filter Response Received: 5 Blacklist Hit Count: 0
Whitelist Hit Count: 0 Reputation Lookup Count: 5 Reputation Action Block: 0 Reputation Action
Pass: 5 Reputation Action Default Pass: 0 Reputation Action Default Block: 0 Reputation Score
None: 0 Reputation Score Out of Range: 0 Category Lookup Count: 5 Category Action Block: 0
Category Action Pass: 5 Category Action Default Pass: 0 Category None: 0 UTM Preprocessor
Internal Statistics ----- Total Packets Received: 193 SSL Packet
Count: 4 Action Drop Flow: 0 Action Reset Session: 0 Action Block: 0 Action Pass: 85 Action
Offload Session: 0 Invalid Action: 0 No UTM Tenant Persona: 0 No UTM Tenant Config: 0 URL Lookup
Response Late: 4 <<<<< Explanation below URL Lookup Response Very Late: 64 <<<<< Explanation
below URL Lookup Response Extremely Late: 2 <<<<< Explanation below Response Does Not Match
Session: 2 <<<<< Explanation below No Response When Freeing Session: 1 First Packet Not From
Initiator: 0 Fail Open Count: 0 Fail Close Count : 0 UTM Preprocessor Internal Global Statistics
----- Domain Filter Whitelist Count: 0 utmdata Used Count:
11 utmdata Free Count: 11 utmdata Unavailable: 0 URL Filter Response Error: 0 No UTM Tenant Map:
0 No URL Filter Configuration : 0 Packet NULL Error : 0 URL Database Internal Statistics -----
----- URL Database Not Ready: 0 Query Successful: 11 Query Successful from
Cloud: 6 <<< 11 queries were succesful but 6 only are queried via brightcloud. 5 (11-6) queries
are cached Query Returned No Data: 0 <<<<<< errors Query Bad Argument: 0 <<<<<< errors Query
Network Error: 0 <<<<<< errors URL Database UTM disconnected: 0 URL Database request failed: 0
URL Database reconnect failed: 0 URL Database request blocked: 0 URL Database control msg
response: 0 URL Database Error Response: 0
===== Files processed:
```

none =====

- «demande tardive » - représente le HTTP GET ou le certificat client/serveur HTTPS [où SNI / DN peut être extrait pour la recherche. La demande en retard est transmise.
- «requêtes très tardives » : signifie qu'une sorte de compteur de perte de session où d'autres paquets dans le flux sont abandonnés jusqu'à ce que le routeur reçoive un verdict d'URL de Brightcloud. En d'autres termes, tout ce qui suit le HTTP GET initial ou le reste du flux SSL sera abandonné jusqu'à réception d'un verdict.
- «requêtes extrêmement tardives » : lorsque la requête de session sur Brightcloud a été réinitialisée sans verdict. La session expire après 60 secondes pour la version < 17.2.1. À partir de 17.2.1, la session d'interrogation vers Brightcloud expirera au bout de 2 secondes. [via [CSCvr98723](#) UTD : Délai d'attente des requêtes URL après deux secondes]

Dans ce scénario, nous voyons des compteurs mondiaux qui mettent en évidence une situation malsaine.

Étape2: Affichage du fichier journal des applications

Le logiciel Unified Thread Detection va enregistrer les événements dans le fichier journal de l'application.

```
cedge6#app-hosting move appid utd log to bootflash:  
Successfully moved tracelog to bootflash:  
iox_utd_R0-0_R0-0.18629_0.20190501005829.bin.gz
```

qui extrait le fichier journal de l'application du conteneur et l'enregistre sur la mémoire Flash elle-même.

L'affichage du journal peut être réalisé à l'aide de la commande suivante :

```
cedge6# more /compressed iox_utd_R0-0_R0-0.18629_0.20190501005829.bin.gz
```

Remarque : dans le logiciel IOS-XE version 20.6.1 et ultérieure, il n'est plus nécessaire de déplacer manuellement le journal des applications UTD. Ces journaux peuvent maintenant être affichés à l'aide de la commande standard **show logging process vman module utd**

L'affichage du journal révèle :

```
.....  
2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 245 , utmdata  
txn_id 0 2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 248 ,  
utmdata txn_id 0 2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id  
249 , utmdata txn_id 0 2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict  
txn_id 250 , utmdata txn_id 0 2020-04-14 17:47:57.660:(#1):SPP-URL-FILTERING txn_id miss match  
verdict txn_id 251 , utmdata txn_id 0 2020-04-14 17:47:57.660:(#1):SPP-URL-FILTERING txn_id miss  
match verdict txn_id 254 , utmdata txn_id 0 2020-04-14 17:47:57.660:(#1):SPP-URL-FILTERING  
txn_id miss match verdict txn_id 255 , utmdata txn_id 0 2020-04-14 17:48:05.725:(#1):SPP-URL-  
FILTERING txn_id miss match verdict txn_id 192 , utmdata txn_id 0 2020-04-14  
17:48:37.629:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 208 , utmdata txn_id 0  
2020-04-14 17:49:55.421:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 211 , utmdata  
txn_id 0 2020-04-14 17:51:40 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection  
timed out 2020-04-14 17:52:21 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection  
timed out 2020-04-14 17:52:21 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection  
timed out 2020-04-14 17:53:56 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
```

```
timed out 2020-04-14 17:54:28 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:54:29 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:54:37 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out
....
```

- « Error : Impossible d'envoyer à l'hôte api.bcti.brightcloud.com » - signifie que la session d'interrogation de Brightcloud a expiré [60 secondes < 17.2.1 / 2 secondes >= 17.2.1]. C'est le signe d'une mauvaise connectivité à Brightcloud. Afin de démontrer le problème, l'utilisation d'EPC [Embedded Packet Capture] permettrait de visualiser le problème de connectivité.
- «SPP-URL-FILTERING txn_id miss match verdict » - Cette condition d'erreur nécessite un peu plus d'explications. La requête Brightcloud est exécutée via un POST où un ID de requête est généré par le routeur

Problème 4

Dans ce scénario, IPS est la seule fonctionnalité de sécurité activée dans UTD, et le client rencontre des problèmes de communication d'imprimante qui est une application TCP.

Dépannage

Pour résoudre ce problème de chemin de données, commencez par capturer les paquets à partir de l'hôte TCP ayant le problème. La capture montre une connexion TCP en trois étapes réussie, mais les paquets de données suivants avec des données TCP semblent avoir été abandonnés par le routeur cEdge. Activez ensuite packet-trace, qui affiche les éléments suivants :

```
edge#show platform packet-trace summ
Pkt   Input           Output           State Reason
0     Gi0/0/1         internal0/0/svc_eng:0  PUNT 64 (Service Engine packet)
1     Tu2000000001   Gi0/0/2         FWD
2     Gi0/0/2         internal0/0/svc_eng:0  PUNT 64 (Service Engine packet)
3     Tu2000000001   Gi0/0/1         FWD
4     Gi0/0/1         internal0/0/svc_eng:0  PUNT 64 (Service Engine packet)
5     Tu2000000001   Gi0/0/2         FWD
6     Gi0/0/1         internal0/0/svc_eng:0  PUNT 64 (Service Engine packet)
7     Tu2000000001   Gi0/0/2         FWD
8     Gi0/0/2         internal0/0/svc_eng:0  PUNT 64 (Service Engine packet)
9     Gi0/0/2         internal0/0/svc_eng:0  PUNT 64 (Service Engine packet)
```

La sortie ci-dessus indique que les paquets 8 et 9 ont été détournés vers le moteur UTD, mais ils n'ont pas été réinjectés dans le chemin de transfert. La vérification des événements de journalisation du moteur UTD ne révèle pas non plus de perte de signature Snort. Vérifiez ensuite les statistiques internes de l'UTD, qui révèlent certaines pertes de paquets dues au normalisateur TCP :

```
edge#show utd engine standard statistics internal
<snip>
Normalizer drops:
    OUTSIDE_PAWS: 0
    AHEAD_PAWS: 0
    NO_TIMESTAMP: 4
    BAD_RST: 0
    REPEAT_SYN: 0
    WIN_TOO_BIG: 0
```

```
WIN_SHUT: 0
BAD_ACK: 0
DATA_CLOSE: 0
DATA_NO_FLAGS: 0
FIN_BEYOND: 0
```

Cause première

La cause première du problème est le comportement incorrect de la pile TCP sur les imprimantes. Lorsque l'option Timestamp est négociée lors de la connexion TCP en 3 étapes, le RFC7323 indique qu'un TCP DOIT envoyer l'option TSopt dans chaque paquet non<RST>. Un examen plus approfondi de la capture de paquets montrera que les paquets de données TCP qui sont abandonnés n'ont pas ces options activées. Avec la mise en oeuvre de l'UTD IOS-XE, le normalisateur Snort TCP avec l'option de bloc est activé indépendamment d'IPS ou d'IDS.

Références

- [Guide de configuration de la sécurité : Défense unifiée contre les menaces](#)