Risoluzione dei problemi di gestione del percorso dati tramite UTD e filtro URL

Sommario

Introduzione

In questo documento viene descritto come risolvere i problemi relativi a Unified Threat Defense (UTD), noto anche come filtro Snort e Uniform Resource Locator (URL) sui router IOS® XE WAN Edge.

Premesse

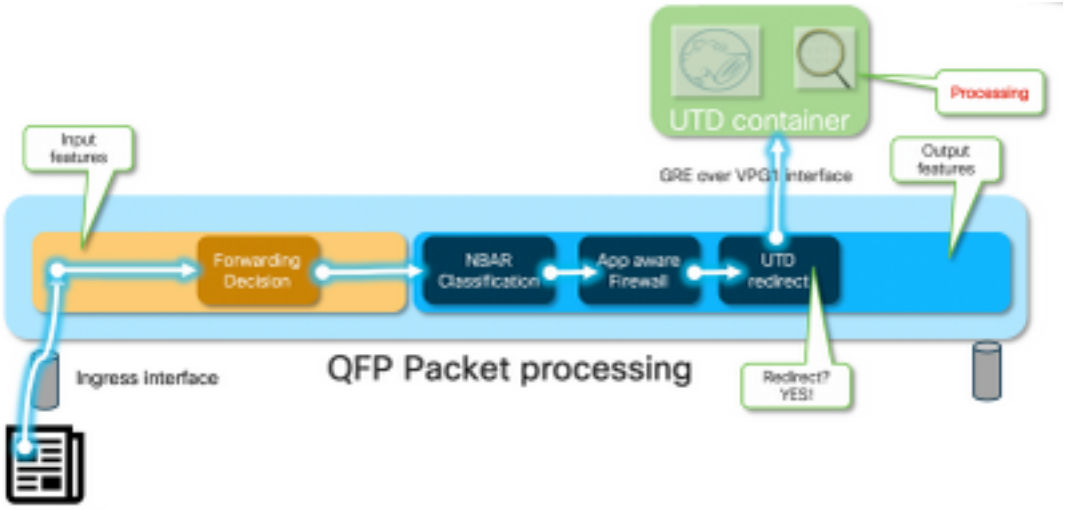
Snort è il sistema di prevenzione delle intrusioni (IPS) più diffuso al mondo. Dal 2013, la società che ha creato una versione commerciale del software Snort, Sourcefire è acquisita da Cisco. A partire dal software 16.10.1 IOS® XE SD-WAN, sono stati aggiunti contenitori di filtro UTD/URF alla soluzione Cisco SD-WAN.

Il contenitore si registra sul router IOS® XE utilizzando il framework app-nav. La spiegazione di questo processo esula dall'ambito del presente documento.

Vista di alto livello di Datapath

Ad alto livello, il percorso dati è simile al seguente:

Dalla LAN/WAN al contenitore



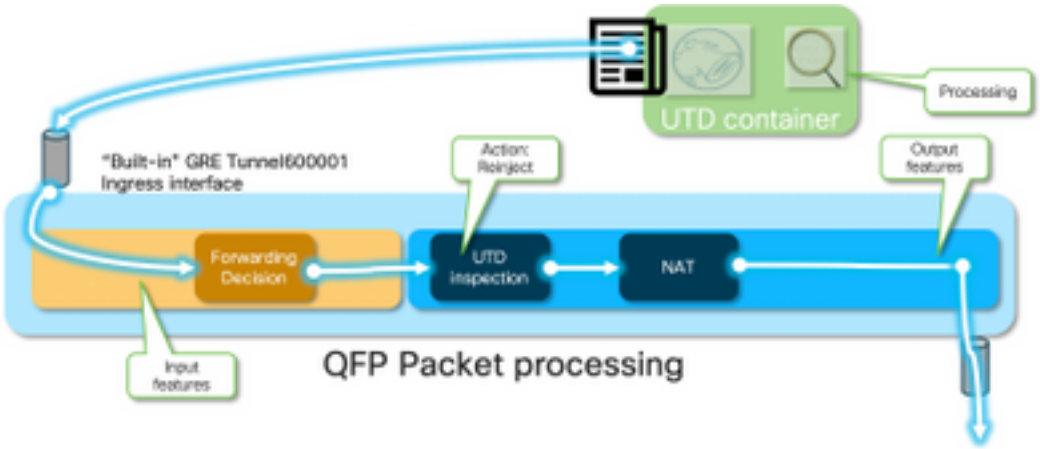
Il traffico proviene dalla LAN. Poiché IOS® XE sa che il contenitore è in uno stato integro, devia il traffico verso il contenitore UTD. La deviazione utilizza l'interfaccia VirtualPortGroup1 come interfaccia in uscita, che incapsula il pacchetto all'interno di un tunnel GRE (Generic Routing Encapsulation).

Il router esegue l'azione "PUNT" sulla causa :64 (pacchetto del motore di servizio) e invia il traffico al processore di routing (RP). Viene aggiunta un'intestazione punt e il pacchetto viene inviato al contenitore utilizzando un'interfaccia di uscita interna verso il contenitore "[internal0/0/svc_eng:0]"

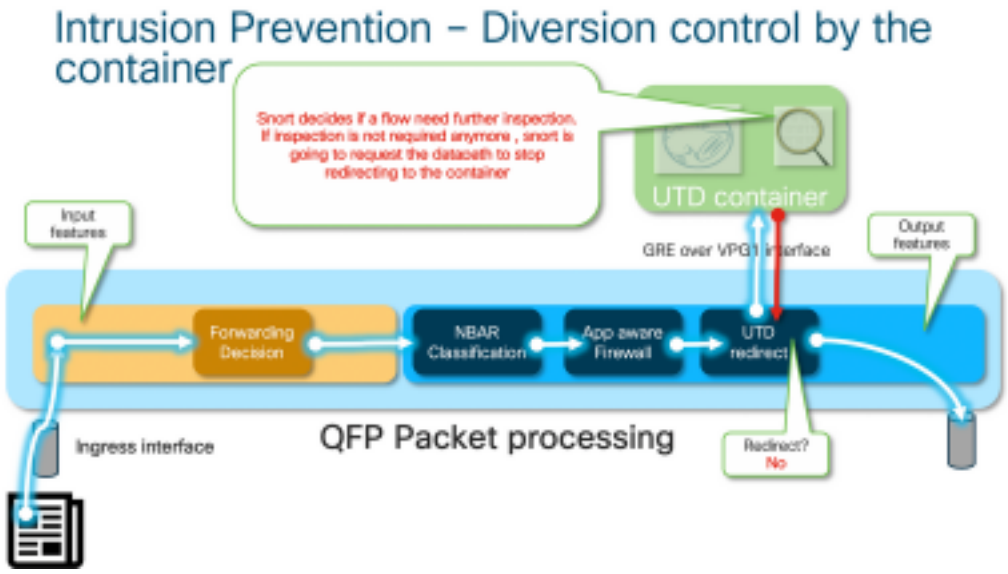
In questa fase, Snort sfrutta i preprocessori e i set di regole. In base ai risultati dell'elaborazione, il pacchetto può essere scartato o inoltrato.

Dal contenitore alla LAN/WAN

Supponendo che il traffico non debba essere scartato, il pacchetto viene inoltrato nuovamente al router dopo l'elaborazione UTD. Sul Quantum Flow Processor (QFP) sembra provenire dal tunnel Tunnel6000001. Quindi, viene elaborato dal router e deve essere (in modo promettente) indirizzato all'interfaccia WAN.



Il contenitore controlla il risultato della diversione nell'ispezione UTD nel percorso dati IOS® XE.



Ad esempio, con il flusso HTTPS, i preprocessori sono interessati a vedere i pacchetti Hello / Hello del server con negoziazione TLS. In seguito, il flusso non viene reindirizzato perché il valore nell'ispezione del traffico crittografato TLS è ridotto.

Datapath Deep Dive

Dal punto di vista del tracer dei pacchetti, queste azioni devono essere verificate (192.168.16.254 è 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
```

Pacchetto in entrata dal lato LAN o WAN verso il contenitore

In questo scenario particolare, il pacchetto tracciato proviene dalla LAN. Dal punto di vista del reindirizzamento, se il flusso proviene da una rete LAN o WAN, esistono differenze rilevanti.

Il client tenta di accedere a www.cisco.com su 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>
```

Il traffico che soddisfa la condizione viene tracciato in entrata sull'interfaccia Gigabit Ethernet2.

```
  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>
```

Sul Feature Invocation Array (FIA) in uscita dell'interfaccia in uscita, UTD FIA ha deciso di deviare questo pacchetto al contenitore.

```
  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>
```

Il pacchetto viene posizionato sul tunnel predefinito Tunnel600001 e viene instradato sull'interfaccia VPG1. In questa fase, il pacchetto originale è incapsulato dal 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
```

Il pacchetto viene trasmesso internamente al contenitore.

> **Nota:** Ulteriori informazioni in questa sezione sui contenitori interni sono fornite solo a scopo informativo. Il contenitore UTD non è accessibile tramite la normale interfaccia CLI.

Andando più a fondo del router, il traffico arriva in un VRF interno sull'interfaccia eth2 del processore di routing:

```
[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 è un'interfaccia TIPC (Transport Inter Process Communication) collegata al processo IOSd. Il canale OneP lo controlla per il passaggio di configurazioni e notifiche tra il contenitore IOSd e UTD.

Dal punto di vista dell'utente, "eth2 [container interface]" è collegato a "VPG1 [192.0.2.1/192.168.2.2]" sono gli indirizzi inviati da vManage a IOS-XE e al contenitore.

Se si esegue **tcpdump**, è possibile vedere il traffico incapsulato del GRE diretto al contenitore. L'incapsulamento GRE include un'intestazione 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.....
```

Pacchetto in entrata dal contenitore verso il lato LAN o WAN

Dopo l'elaborazione Snort (presupponendo che il traffico non venga scartato), viene reinserito nel percorso di inoltro QFP.

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

Tunnel60001 è l'interfaccia in uscita dal contenitore.

```
  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
```

Poiché il traffico è già stato ispezionato, il router sa che si tratta di una reiniezione.

```
  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
```

Il traffico arriva al NAT e va verso Internet.

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

Integrazione registrazione flusso UTD con Packet-trace

IOS-XE 17.5.1 ha aggiunto l'integrazione della registrazione di flusso UTD con packet-trace, dove l'output della traccia del percorso includerà un verdetto UTD. Il valore Verdict può essere uno dei seguenti, ad esempio:

- il pacchetto che UTD decide di bloccare/avvisare per Snort
- consenti/rilascia per URLF
- blocca/consenti AMP

Per i pacchetti che non dispongono di informazioni sul verdetto UTD, non vengono registrate informazioni sulla registrazione del flusso. Si noti inoltre che non è disponibile alcuna registrazione di IPS/IDS pass/allow verdict a causa del potenziale impatto negativo sulle prestazioni.

Per abilitare l'integrazione del log di flusso, utilizzare il modello aggiuntivo CLI con:

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

Output di esempio per verdetti diversi:

Timeout ricerca 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)
```

La reputazione e il verdetto dell'URLF permettono:

```
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
```

Blocco della reputazione e del verdetto dell'URLF:

```
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
```

Prerequisito:

Verifica della compatibilità della versione UTD con 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)
```

Se viene visualizzato "UNSUPPORTED", l'aggiornamento del contenitore è richiesto come primo passo prima di iniziare la risoluzione dei problemi.

Verifica la presenza di una configurazione valida del server dei nomi nel contenitore

Alcuni servizi di sicurezza, ad esempio AMP e URLF, richiedono che il contenitore UTD sia in grado di risolvere i nomi dei provider di servizi cloud, pertanto il contenitore UTD deve disporre di configurazioni del server dei nomi valide. È possibile verificare questa condizione verificando il file resolv.conf per il contenitore nella shell di sistema:

```
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
```

Problema 1

In base alla progettazione, Unified Thread Defense deve essere configurato insieme a Direct Internet Access Use Case (DIA). Il contenitore tenterà di risolvere **api.bcti.brightcloud.com** per

interrogare la reputazione e le categorie degli URL. Nell'esempio, nessuno degli URL ispezionati viene bloccato anche se viene applicata la configurazione corretta

Risoluzione dei problemi

Esaminare sempre il file di registro del contenitore.

```
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
```

che copia il file di registro sul flash stesso.

Per visualizzare il registro, usare il comando:

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

La visualizzazione del registro rivela:

```
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
```

Per impostazione predefinita, vManage esegue il provisioning di un contenitore che utilizza il server OpenDNS [208.67.222.222 e 208.67.220.220]

Causa principale

Il traffico DNS (Domain Name System) per risolvere **api.bcti.brightcloud.com** viene interrotto nel percorso tra il contenitore e i server DNS ombrello. Verificare sempre che entrambi i DNS siano raggiungibili.

Problema 2

In uno scenario in cui i siti Web delle categorie Computer e Informazioni su Internet dovrebbero essere bloccati, la richiesta HTTP a www.cisco.com viene correttamente eliminata mentre le richieste HTTPS non lo sono.

Risoluzione dei problemi

Come spiegato prima, il traffico viene punito al container. Quando questo flusso è incapsulato nell'intestazione GRE, il software aggiunge anche un'intestazione VPATH. Sfruttando questa intestazione, il sistema consente di passare una condizione di debug al contenitore stesso. Ciò significa che i contenitori UTD sono facilmente gestibili.

In questo scenario, l'indirizzo IP del client è 192.168,16.254. Vediamo come risolvere i problemi relativi alla gestione degli snort da parte del contenitore stesso per il traffico proveniente dal client.

```
debug platform condition ipv4 192.168.16.254/32 both
debug platform condition feature utd controlplane submode serviceplane-web-filtering level
verbose
debug platform condition start
```

Questo gruppo di comandi indica a IOS-XE di contrassegnare il traffico da o verso 192.168.16.254. Ciò consente di passare il flag debug-me al contenitore tramite l'intestazione VPATH

LSMPI punt header	Outer IP header (e.g. 192.0.2.x)	GRE header	vPath header (conditional debug flag is here)	Inner (original) IP packet

Il debug degli snort viene eseguito solo su quel flusso specifico, mentre gli altri vengono elaborati normalmente.

In questa fase, è possibile chiedere all'utente di attivare il traffico dal client verso www.cisco.com.

Il passaggio successivo consiste nel recuperare i log:

```
app-hosting move appid utd log to bootflash:
```

Nel caso del traffico HTTP, il preprocessore HTTP snort individua l'URL nella richiesta get.

```
2019-04-26 13:04:27.773:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 39540, p->dst_port
= 80
2019-04-26 13:04:27.793:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 80, p->dst_port =
39540
2019-04-26 13:04:27.794:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 39540, p->dst_port
= 80
2019-04-26 13:04:27.794:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 39540, p->dst_port
= 80
2019-04-26 13:04:27.794:(#1):SPP-URL-FILTERING got utmdata_p
2019-04-26 13:04:27.794:(#1):SPP-URL-FILTERING HTTP Callback, direction = 00000080

2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING White list regex match not enabled
2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING Black list regex match not enabled
2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING URL database Request: url_len = 12, msg overhead
12 url: www.cisco.com/ <<<<<<<
2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING Send to URL database: req_id=0x10480047
2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING Sent to URL database 24 bytes
2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING Send to URL database done, idx: 71, URL:
www.cisco.com/
2019-04-26 13:04:27.795:(#1):SPP-URL-FILTERING Received from URL database 24 bytes
2019-04-26 13:04:27.816:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 80, p->dst_port =
39540
2019-04-26 13:04:27.816:(#1):SPP-URL-FILTERING Found UTMData at 0x007f8d9ee80878, action =
0000000a
2019-04-26 13:04:27.816:(#1):SPP-URL-FILTERING Utm_verdictProcess: vrf_id 1, category 0x63,
score 81 <<<<<<<<<<<<<<<<<<<<<<<<<
2019-04-26 13:04:27.816:(#1):SPP-URL-FILTERING Category 0x3f
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
2019-04-26 13:04:27.816:(#1):SPP-URL-FILTERING index = 63, action = 1
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
2019-04-26 13:04:27.816:(#1):SPP-URL-FILTERING Blocking category = 0x3f
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

In caso di traffico HTTPS, il DNS di destinazione è stato estratto dal preprocessore HTTPS dal server hello

```
2019-05-01 00:56:18.870:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 35322, p->dst_port
= 443
2019-05-01 00:56:18.886:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.887:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 35322, p->dst_port
= 443
2019-05-01 00:56:18.887:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 35322, p->dst_port
= 443
2019-05-01 00:56:18.903:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.906:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.906:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 35322, p->dst_port
= 443
2019-05-01 00:56:18.907:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.907:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.907:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322

2019-05-01 00:56:18.908:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.908:(#1):SPP-URL-FILTERING utm_sslLookupCallback
2019-05-01 00:56:18.908:(#1):SPP-URL-FILTERING got utmdata_p
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING White list regex match not enabled
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING Black list regex match not enabled
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING URL database Request: url_len = 11, msg overhead
12 url: www.cisco.com  <<<<<<<<
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING Send to URL database: req_id=0x10130012
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING Sent to URL database 23 bytes
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING Send to URL database done, idx: 18, URL:
www.cisco.com
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322
2019-05-01 00:56:18.910:(#1):SPP-URL-FILTERING Found UTMData at 0x007f1d9c479640, action =
00000008
2019-05-01 00:56:18.910:(#1):SPP-URL-FILTERING Verdict very late, in queryig state 2, idx=18
2019-05-01 00:56:18.910:(#1):SPP-URL-FILTERING UTM preprocessor p->src_port = 443, p->dst_port =
35322

2019-05-01 00:56:18.910:(#1):SPP-URL-FILTERING Found UTMData at 0x007f1d9c479640, action =
00000009
2019-05-01 00:56:18.910:(#1):SPP-URL-FILTERING Verdict very late, in queryig state 2, idx=18
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
2019-05-01 00:56:18.909:(#1):SPP-URL-FILTERING Received from URL database 24 bytes
```

In questa schermata non viene attivata la pagina di blocco poiché il software non riporta i risultati della query Webroot.

Causa principale

CSCvo7664 "Il filtro URL UTD per la ricerca di categorie non è riuscito con un errore di ricerca tramite Web" si riferisce alla perdita di traffico quando il software non ha ancora risposto alla nostra richiesta di verdetto URL.

Problema 3

In questo scenario, vengono interrotte a intermittenza le sessioni di esplorazione Web che dovrebbero essere consentite dal filtro URL [a causa della relativa classificazione]. Ad esempio, l'accesso a www.google.com non è casuale anche se è consentita la categoria "motore di ricerca Web".

Risoluzione dei problemi

Fase 1: Raccolta delle statistiche generali

> **Nota** : questo output del comando viene reimpostato ogni 5 minuti

```
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 ================================================================================
```

- "late request": rappresenta il GET HTTP o il certificato client/server HTTPS [dove è possibile estrarre SNI / DN per la ricerca. Le richieste in ritardo vengono inoltrate.

- "richieste molto in ritardo" - indica che una sorta di contatore di perdita di sessione in cui ulteriori pacchetti nel flusso vengono scartati finché il router non riceve un verdetto URL da Brightcloud. In altre parole, qualsiasi elemento successivo all'HTTP GET iniziale o al flusso SSL rimanente verrà eliminato fino alla ricezione di un verdetto.
- "richieste estremamente in ritardo" - quando la sessione di query su Brightcloud è stata reimpostata senza fornire un verdetto. La sessione scadrà dopo 60 secondi per la versione < 17.2.1. A partire dalla versione 17.2.1, la sessione di query su Brightcloud scadrà dopo 2 secondi. [tramite CSCvr98723 UTD: Timeout [richieste URL dopo due secondi]

In questo scenario, vediamo contatori globali che evidenziano una situazione dannosa.

Fase 2: Visualizzazione del file di registro dell'applicazione

Il software Unified Thread Detection registrerà gli eventi nel file registro dell'applicazione.

```
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
```

che estrae il file di registro dell'applicazione contenitore e lo salva sul flash stesso.

Per visualizzare il registro, usare il comando:

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

> **Nota:** nel software IOS-XE versione 20.6.1 e successive non è più necessario spostare manualmente il registro applicazioni UTD. È ora possibile visualizzare questi log utilizzando il comando standard **show log process vman module utd**

La visualizzazione del registro rivela:

```
.....
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
....
```

- "ERRORE: "Impossibile inviare all'host api.bcti.brightcloud.com" - indica che la sessione di query su Brightcloud è scaduta [60 secondi < 17.2.1 / 2 secondi >= 17.2.1]. Questo è il

segno di una cattiva connettività a Brightcloud.
Per dimostrare il problema, l'uso di EPC [Embedded Packet Capture] consente di visualizzare il problema di connettività.

- "SPP-URL-FILTERING txn_id miss match verdict" - Questa condizione di errore richiede una spiegazione più dettagliata. La query Brightcloud viene eseguita tramite un POST in cui il router genera un ID query

Problema 4

In questo scenario, IPS è l'unica funzionalità di sicurezza abilitata in UTD e il cliente ha problemi con la comunicazione con la stampante che è un'applicazione TCP.

Risoluzione dei problemi

Per risolvere il problema del percorso dati, acquisire il pacchetto dall'host TCP che ha il problema. L'acquisizione mostra un handshake TCP a 3 vie riuscito, ma i successivi pacchetti di dati con dati TCP sembrano essere stati scartati dal router cEdge. Quindi abilitare packet-trace, che ha mostrato quanto segue:

```
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)
```

L'output sopra riportato indica che i pacchetti numero 8 e 9 sono stati deviati al motore UTD ma non sono stati reinseriti nel percorso di inoltro. Il controllo degli eventi di registrazione del motore UTD inoltre non rivela alcuna perdita di firma Snort. Controllare quindi le statistiche interne dell'UTD, che rivelano alcune perdite di pacchetti dovute al normalizzatore 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
```

Causa principale

La causa principale del problema è un comportamento errato dello stack TCP sulle stampanti. Quando l'opzione Timestamp (Data e ora) viene negoziata durante l'handshake a 3 vie TCP, la RFC 7323 indica che TCP DEVE inviare l'opzione TSopt in ogni pacchetto non <RST>. Un esame più attento dell'acquisizione dei pacchetti mostrerà che i pacchetti di dati TCP che vengono scartati non hanno queste opzioni abilitate. Con l'implementazione UTD di IOS-XE, Snort TCP normalizer con l'opzione block è abilitato indipendentemente da IPS o IDS.

Riferimenti

- Guida alla configurazione della protezione: Unified Threat Defense