Probleemoplossing voor gegevensverwerking via UTD en URL-filtering

Inhoud

[Inleiding](#)
[Achtergrondinformatie](#)
[Datapath-weergave op hoog niveau](#)
[Van LAN/WAN naar de container](#)
[Van container tot LAN/WAN](#)
[Datapath Deep Dive](#)
[IP-pakketten van LAN of WAN naar de containerzijde](#)
[Indrukpakket van containerlijn naar LAN of WAN-zijde](#)
[Integratie van UTD-stroming met Packet-sporen](#)
[Voorlopig:](#)
[Controleer of de UTD-versie compatibel is met IOS XE](#)
[Controleer op geldige configuratie van de nameserver in de houder](#)
[Probleem 1](#)
[Problemen oplossen](#)
[Root-oorzaak](#)
[Probleem 2](#)
[Problemen oplossen](#)
[Root-oorzaak](#)
[Probleem 3](#)
[Problemen oplossen](#)
[Stap 1: Algemene statistieken verzamelen](#)
[Stap 2: Het logbestand van de toepassing bekijken](#)
[Probleem 4](#)
[Problemen oplossen](#)
[Root-oorzaak](#)
[Referenties](#)

Inleiding

Dit document beschrijft hoe u Unified Threat Defense (UTD), ook bekend als URL-filtering van SNEL en Uniform Resource Locator (URL) op IOS XE WAN-routers, kunt oplossen.

Achtergrondinformatie

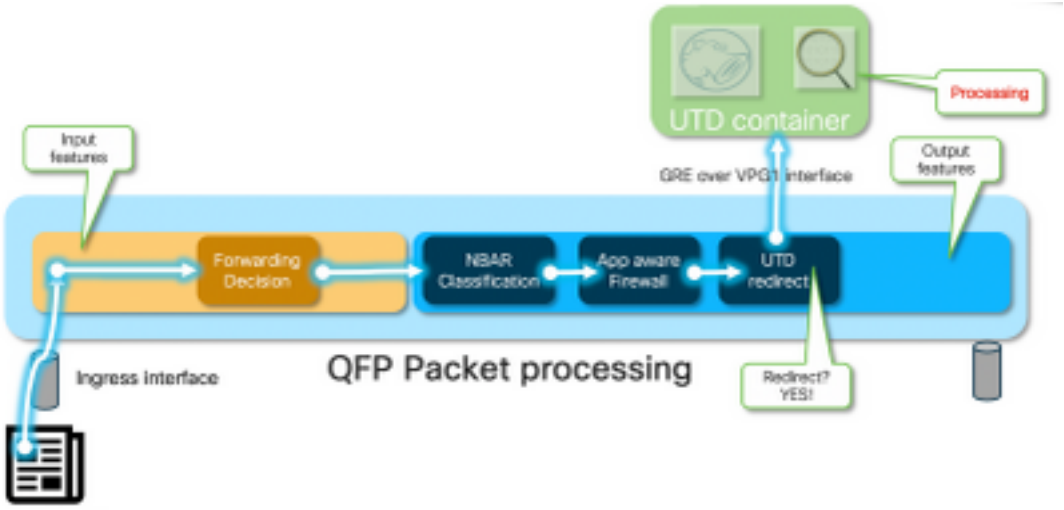
Snort is het meest gebruikte Inbraakpreventiesysteem (IPS) ter wereld. Sinds 2013 heeft Sourcefire, het bedrijf dat een commerciële versie van de kortesoftware heeft gemaakt, overgenomen door Cisco. Vanaf 16.10.1 IOS® XE SD-WAN software zijn UTD/URF-Filtering containers toegevoegd aan de Cisco SD-WAN oplossing.

De container registreert aan de IOS®XE router door het app-nav kader te gebruiken. De uitleg van dit proces valt buiten het toepassingsgebied van dit document.

Datapath-weergave op hoog niveau

Op een hoog niveau ziet de datapath er zo uit:

Van LAN/WAN naar de container



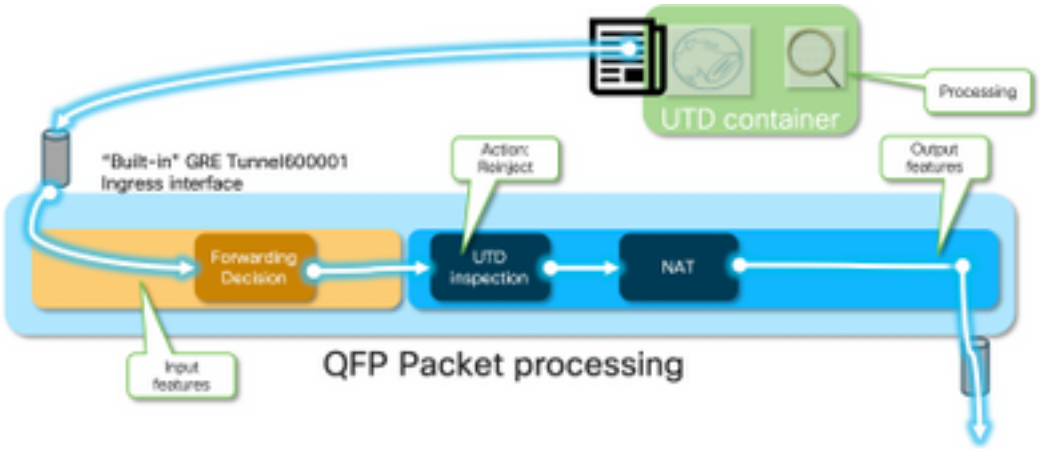
Het verkeer komt van de LAN kant. Omdat IOS® XE weet dat de container in een gezonde toestand is, leidt het het verkeer naar de UTD-container. De afleiding gebruikt de interface VirtualPortGroup1 als de ress interface, die het pakket in een Generic Routing Encapsulation-tunnel (GRE) inkapselt.

De router voert "PUNT" actie uit met oorzaak 4.64 (het pakket van de Services Engine)" en stuurt het verkeer naar de routeprocessor (RP). Er wordt een puntkop toegevoegd en het pakket wordt naar de container verzonden met een interne opening naar de container "[interne 0/0/svc_eng:0"

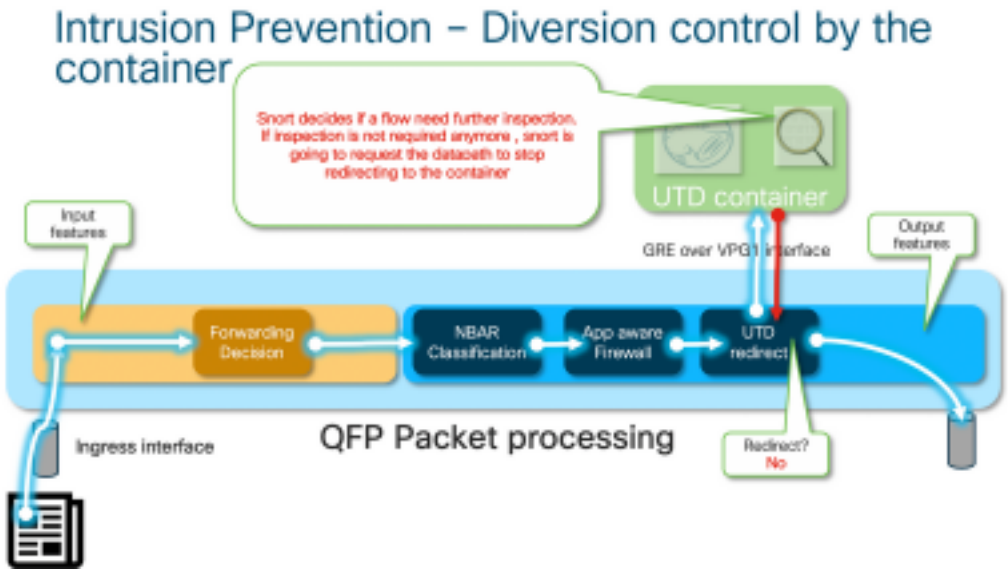
In dit stadium maakt Snort gebruik van zijn voorprocessors en regeltjes. U kunt het pakket op basis van de verwerkingsresultaten laten vallen of doorsturen.

Van container tot LAN/WAN

aangenomen dat het verkeer niet zou moeten worden ingetrokken, wordt het pakket teruggestuurd naar de router na UTD-verwerking. Het komt van Tunnel 600001 op de Quantum Flow Processor (QFP). Daarna wordt het verwerkt door de router en moet (hopelijk) naar de WAN-interface worden geleid.



container controleert het afleidingsresultaat in de UTD-inspectie in de IOS® XE datapath.



Bijvoorbeeld, met HTTPS flow, zijn de preprocessoren geïnteresseerd om de server Hallo/Client Hallo-pakketten met TLS-onderhandeling te zien. Daarna wordt de stroom niet omgeleid omdat er weinig waarde is voor het inspecteren van het gecodeerde TLS-verkeer.

Datapath Deep Dive

Vanuit een pakkettracer-standpunt zullen deze reeks handelingen worden gezien (192.168.16.254 is een webclient):

```
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
```

IP-pakketten van LAN of WAN naar de containerzijde

In dit specifieke scenario, komt het getraceerde pakket van LAN. Vanuit een omleidingsstandpunt zijn er relevante verschillen als de stroom uit LAN of WAN komt.

De client probeert toegang te krijgen tot www.cisco.com voor 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>
```

Verkeer dat de conditie wordt aangepast is getraceerd toegang op interface 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>
```

Op de ress Feature Invocation Array (FIA) van de spanning-interface heeft UTD FIA besloten dit pakje naar de houder te sturen.

```
  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>
```

Het pakket wordt op de standaardtunnel Tunnel600001 geplaatst en wordt over de VPG1 interface routeerd. In deze fase is het oorspronkelijke pakket GRE ingekapseld.

```
  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
```

Het pakket wordt intern naar de container verzonden.

> **Opmerking:** Verdere informatie in deze rubriek over de interne containers wordt uitsluitend ter informatie verstrekt. De UTD-container is niet toegankelijk via de normale CLI-interface.

Hoe dieper in de router zelf, het verkeer komt in een interne VRF aan op de interface van de routeprocessor eth2:

```
[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 is een interface voor transportcommunicatie (TIPC) die is aangesloten op het IOS-proces. Het OneP-kanaal voert het uit voor het doorgeven van configuraties en kennisgevingen heen en weer tussen de IOSd en UTD-container.

Op basis van wat u bezorgd bent, is "eth2 [containerinterface]" overbrugd naar "VPG1 [192.0.2.1/192.168.2.2]" de adressen die door vManager naar IOS-XE en container worden geduwd.

Als je **tcPDump** gebruikt, zie je dat het ingekapselde GRE-verkeer naar de container gaat. De GRE-insluiting bevat een VPATH-header.

```
[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.....
```

Indrukpakket van containerlijn naar LAN of WAN-zijde

Na de verwerking van de snor (ervan uitgaande dat het verkeer niet mag worden laten vallen) wordt het opnieuw in het QFP-verzendpad geplaatst.

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

Tunnel600001 is de spanning-interface van de container.

```
  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
```

Aangezien het verkeer al geïnspecteerd is, weet de router dat dit een herinjectie is.

```
  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
```

Het verkeer krijgt NATed en gaat naar het internet.

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

Integratie van UTD-stroming met Packet-sporen

IOS-XE 17.5.1 toegevoegde UTD-debietlogingintegratie met pakketsporen, waar de pad-sporenuitvoer een UTD-oordeel bevat. Een uitspraak kan bijvoorbeeld het volgende zijn:

- het pakket dat UTD besluit te blokkeren/te waarschuwen voor snort
- Laat/drop voor URLF
- blokkeren/toelaten van AMP

Voor pakketten die niet de UTD oordeel informatie hebben, wordt geen informatie van het stroomloggen geregistreerd. Merk ook op dat er geen registratie is van IPS/IDS-passeren/oordelen door mogelijk negatieve prestatiekortingen.

Gebruik de CLI Add-on sjabloon met:

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

Uitvoer van voorbeeld voor verschillende oordelen:

Time-out bij URL-opname:

```
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)
```

reputatie en vonnis van URLF staat toe:

```
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
```

Blokkenfunctie van URLF en vonnis:

```
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
```

Voorlopig:

Controleer of de UTD-versie compatibel is met 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)
```

Als "ONONDERSTEUND" wordt weergegeven, is de containerupgrade vereist als eerste stap voordat u een oplossing begint.

Controleer op geldige configuratie van de nameserver in de houder

Sommige beveiligingsservices zoals AMP en URLF vereisen dat de UTD-container namen voor de cloudservice-providers kan oplossen, dus de UTD-houder moet beschikken over geldige nameserverconfiguraties. Dit kan worden geverifieerd door het bestand resolv.conf voor de container onder de systeemschelp te controleren:

```
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
```

Probleem 1

Per ontwerp moet de Unified Thread Defense in zijn geheel worden geconfigureerd met de Direct Internet Access Use Case (DIA). De container zal proberen om **api.bcti.heldercloud.com** op te

lossen om reputaties en categorieën van URL af te vragen. In dit voorbeeld wordt geen van de geïnspecteerde URL's geblokkeerd zelfs als de juiste configuratie wordt toegepast

Problemen oplossen

Kijk altijd naar het containerlogbestand.

```
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
```

Dat kopieert het logbestand op de flitser zelf.

U kunt met deze opdracht het logbestand weergeven:

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

Het logbestand toont aan:

```
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
```

Standaard vManager-bepalingen is een container die OpenDNS-server gebruikt [208.67.222.222 en 208.67.220.220]

Root-oorzaak

Domain Name System (DNS)-verkeer om **api.bcti.heldercloud.com** op te lossen wordt ergens in het pad tussen de container en de paraplu DNS-servers gedropt. Zorg er altijd voor dat beide DNS bereikbaar zijn.

Probleem 2

In een scenario waarin websites van de categorieën Computer- en Internet Info geblokkeerd zouden worden, wordt het verzoek om http naar www.cisco.com correct ingetrokken terwijl er geen aanvragen voor HTTPS zijn.

Problemen oplossen

Zoals eerder is uitgelegd, wordt het verkeer naar de container geduwd. Wanneer deze flow is opgenomen in de GRE-header, wordt ook software toegevoegd en een VPATH-header toegevoegd. Met behulp van deze header kan het systeem een debug-conditie aan de container zelf doorgeven. Dit betekent dat UTD-containers goed bruikbaar zijn.

In dit scenario is het IP-adres van de client 192.168.16.254. Laten we problemen oplossen met de snort-verwerking door de container zelf voor het verkeer dat van mijn cliënt komt.

```
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
```

Deze reeks opdrachten geeft IOS-XE instructies om verkeer te markeren van of naar 192.168.16.254. Hiermee kan debug-me vlag worden doorgegeven naar de container via de VPATH header

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

Snort debugs alleen die specifieke stroom terwijl anderen normaal worden verwerkt.

In dit stadium kunt u de gebruiker vragen om het verkeer van de client naar www.cisco.com te activeren.

De volgende stap is het ophalen van de logbestanden:

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

In het geval van HTTP verkeer, ontdekt de gesnate HTTP preprocessor de URL in de krijg aanvraag.

```
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 het geval van https-verkeer, is de doelDNS van de server-hallo verwijderd door de HTTPS-voorprocessor

```
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
```

Hier ziet u niet dat de blokkerende pagina wordt geactiveerd omdat de software de resultaten van de webroot query niet rapporteert.

Root-oorzaak

CSCvo7664 "UTD URL filteren voor categorie lookup faalt bij zoekoproerzoekstoring" gaat over het verkeer dat wordt uitgelekt als de software nog geen reactie heeft op ons verzoek om een

URL-vonnis.

Probleem 3

In dit scenario wordt website browsing-sessies die door URL-filtering [vanwege hun classificatie] toegestaan zouden moeten zijn, met tussenpozen verbroken. Bijvoorbeeld, toegang tot www.google.com is willekeurig niet mogelijk zelfs als de categorie "web search engine" is toegestaan.

Problemen oplossen

Stap 1: Algemene statistieken verzamelen

> **Opmerking** Deze opdrachtoutput wordt om de 5 minuten teruggezet

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

- "vertraagd verzoek" - vertegenwoordigt HTTP GET of het HTTPS client/server certificaat [

waar SNI / DN kan worden geëxtraheerd voor raadpleging . Een te laat verzoek wordt toegezonden.

- "zeer late verzoeken" - betekent dat een of ander soort sessie teller bevat waar verdere pakketten in de stroom worden verzonden tot de router een URL vonnis van Brightcloud ontvangt. Met andere woorden: alles na de eerste HTTP GET en de resterende SSL flow zal gezakt worden tot een oordeel is ontvangen.
- "extreem late verzoeken" - wanneer de sessie query naar Brightcloud is gereset zonder een oordeel te geven. De sessie zal na 60 seconden worden uitgesteld voor versie < 17.2.1. Vanaf 17.2.1 zal de zoeksessie naar Brightcloud na 2 seconden worden uitgesteld. [via CSCvr98723 UTD: Time-out URL-verzoeken na twee seconden]

In dit scenario zien we mondiale tegenstellingen die de nadruk leggen op een ongezonde situatie.

Stap 2: Het logbestand van de toepassing bekijken

De software voor Unified Thread Detectie zal gebeurtenissen in het logbestand van de toepassing opnemen.

```
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
```

Dat haalt het logbestand van de containertoepassing uit en slaat het op de flitser zelf op.

U kunt met deze opdracht het logbestand weergeven:

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

> **Opmerking:** In IOS-XE softwarerelease 20.6.1 en hoger is het niet langer vereist om het UTD-toepassingslogbestand handmatig te verplaatsen. Deze logbestanden kunnen nu worden bekeken met behulp van de standaard opdracht **om het logproces vman-module weer te geven**

Het logbestand toont aan:

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

- "FOUT: Kan niet naar host api.bcti.brightcloud.com" sturen - betekent dat de zich voordoende sessie naar Heldercloud de timing van [60 seconden < 17.2.1 / 2 seconden >= 17.2.1] is geweest. Dit is het teken van een slechte verbinding met de Helderwolk.
  Om dit probleem aan te tonen, zou het gebruik van EPC [Embedded Packet Capture] het aansluitingsprobleem kunnen visualiseren.
- "SPP-URL-FILTERING txn_id missen match oordeel" - Deze foutvoorwaarde vereist een beetje meer uitleg. Brightcloud query wordt uitgevoerd via een POST waar een query-ID wordt gegenereerd door de router

Probleem 4

In dit scenario is IPS de enige veiligheidsfunctie die in UTD wordt ingeschakeld en de klant ervaart problemen met printercommunicatie die een TCP-toepassing is.

Problemen oplossen

Om deze datapath-kwestie op te lossen, neem eerst de pakketopname van de TCP-host met de probleem. De opname toont een succesvolle TCP 3-handdruk, maar de volgende gegevenspakketten met TCP-gegevens lijken te zijn gevallen door de cEdge-router. Stel vervolgens pakketsporen in, hetgeen het volgende heeft aangetoond:

```
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)
```

De bovenstaande uitvoer geeft pakketnummer 8 en 9 aan en zijn naar de UTD-motor omgeleid, maar zijn niet opnieuw geïnjecteerd in het verzendingspad. Controleer of de UTD-motorhoutkap iets anders is dan het registreren van handtekeningen door een snort. Controleer vervolgens de interne statistieken van UTD, die wat pakketdalingen door de TCP-normalisatie zichtbaar maken:

```
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
```

Root-oorzaak

De oorzaak van het probleem is te wijten aan verkeerd gedraaid TCP-stack op de printers. Wanneer de optie Time-stamp tijdens de TCP 3-handdruk is onderhandeld, verklaart RFC7323 dat TCP de TSopt-optie in elk niet-<RST>-pakket moet verzenden. Een nauwer onderzoek van de pakketvastlegging zal tonen dat de TCP-datapakketten die worden geworpen niet deze opties hebben ingeschakeld. Met de IOS-XE UTD implementatie, wordt de normale TCP-normalizer van de SNEL met de blokoptie geactiveerd ongeacht IPS of IDS.

Referenties

- Security configuratiegids: Unified Threat Defense