Fehlerbehebung bei DataPath-Behandlung durch UTD und URL-Filterung

Inhalt

Einleitung

In diesem Dokument wird die Fehlerbehebung für Unified Threat Defense (UTD), auch bekannt als Snort and Uniform Resource Locator (URL) Filtering auf IOS® XE WAN Edges Routern, beschrieben.

Hintergrundinformationen

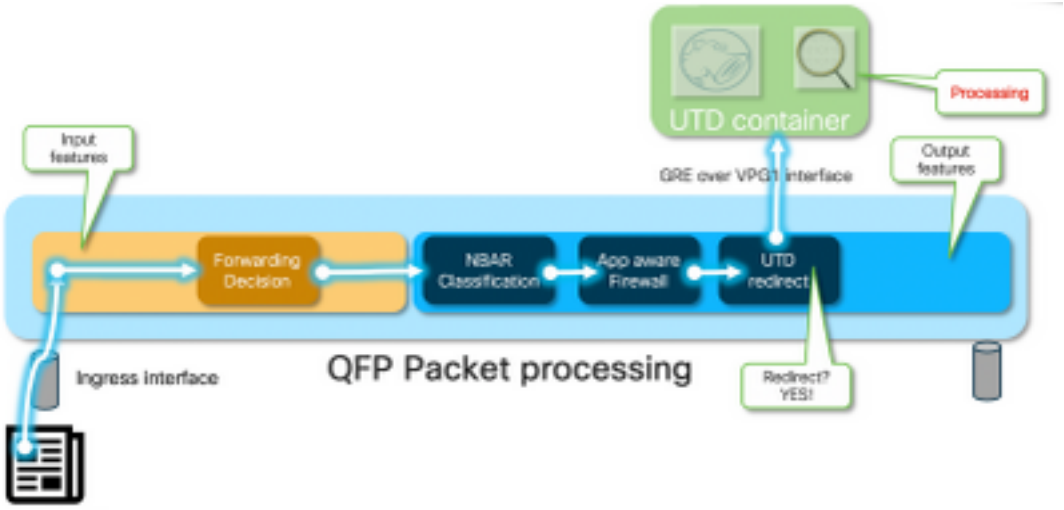
Snort ist das weltweit am häufigsten eingesetzte Intrusion Prevention System (IPS). Seit 2013 wird Sourcefire von Cisco übernommen, dem Unternehmen, das eine kommerzielle Version der Snort-Software erstellt hat. Ab der Version 16.10.1 der SD-WAN-Software IOS® XE wurden der Cisco SD-WAN-Lösung UTD/URF-Filterungscontainer hinzugefügt.

Der Container wird mithilfe des app-nav-Frameworks beim IOS®XE-Router registriert. Die Erläuterung dieses Prozesses geht über den Rahmen dieses Dokuments hinaus.

Datapath - High-Level-Ansicht

Allgemein sieht der Datenpfad wie folgt aus:

Vom LAN/WAN zum Container



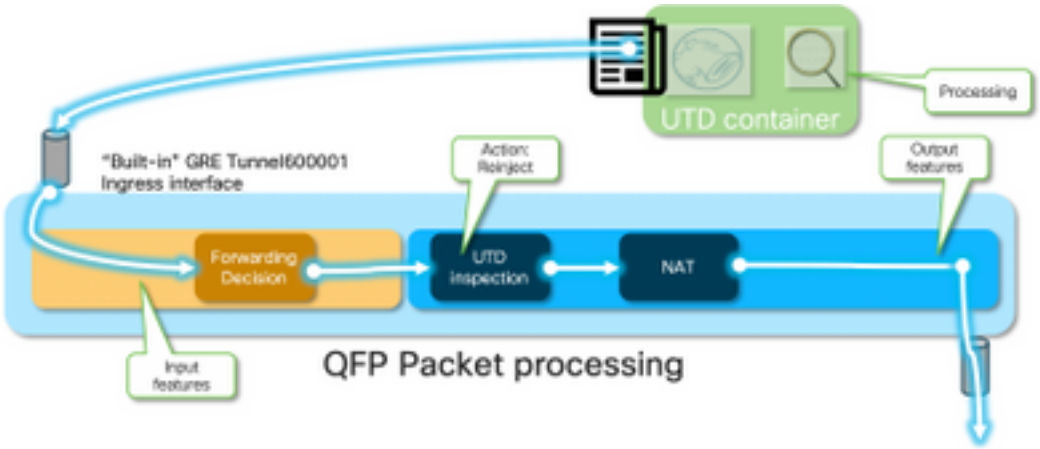
Datenverkehr kommt von der LAN-Seite. Da IOS® XE weiß, dass sich der Container in einem einwandfreien Zustand befindet, leitet er den Datenverkehr an den UTD-Container um. Bei der Umleitung wird die VirtualPortGroup1-Schnittstelle als Ausgangsschnittstelle verwendet, die das Paket in einen GRE-Tunnel (Generic Routing Encapsulation) kapselt.

Der Router führt eine PUNT-Aktion mit der Ursache :64 (Service-Engine-Paket) durch und sendet den Datenverkehr an den Routingprozessor (RP). Es wird ein Stack-Header hinzugefügt, und das Paket wird mithilfe einer internen Ausgangsschnittstelle zum Container "[internal0/0/svc_eng:0]" an den Container gesendet.

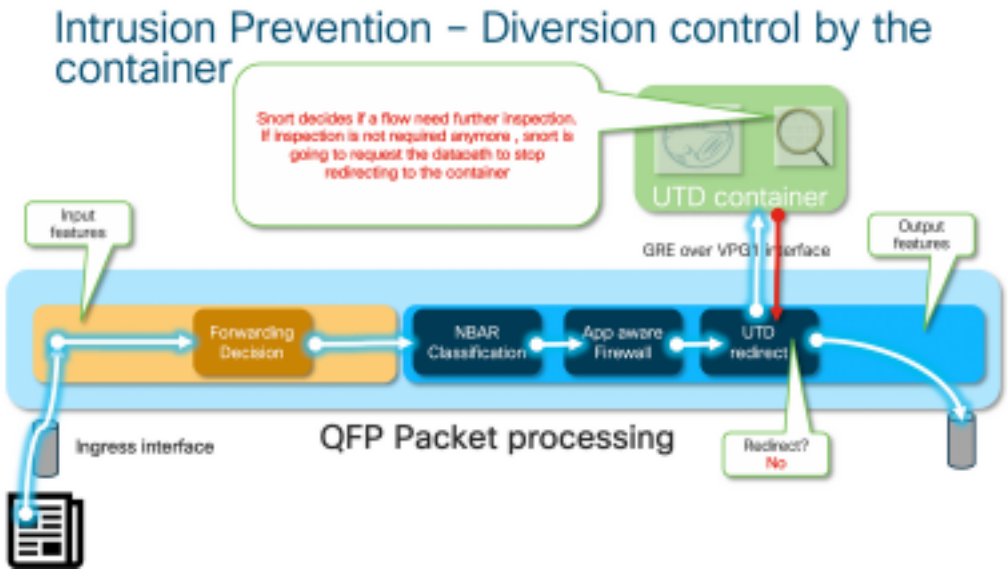
In dieser Phase nutzt Snort seine Präprozessoren und Regelsätze. Das Paket kann basierend auf den Verarbeitungsergebnissen verworfen oder weitergeleitet werden.

Vom Container zum LAN/WAN

Wenn der Datenverkehr nicht verworfen werden soll, wird das Paket nach UTD-Verarbeitung an den Router zurückgeleitet. Es scheint auf dem Quantum Flow Processor (QFP) von Tunnel600001 zu stammen. Anschließend wird sie vom Router verarbeitet und muss (hoffentlich) zur WAN-Schnittstelle geroutet werden.



Container steuert die Umleitung, was zur UTD-Prüfung im IOS® XE-Datenpfad führt.



Beispielsweise sind die Präprozessoren bei HTTPS-Fluss daran interessiert, die Hello-/Client Hello-Pakete des Servers mit TLS-Aushandlung zu sehen. Anschließend wird der Datenfluss nicht umgeleitet, da die Überprüfung von TLS-verschlüsseltem Datenverkehr wenig sinnvoll ist.

Datapath im Detail

Aus Sicht des Packet-Tracer wird dieser Aktionssatz sichtbar sein (192.168.16.254 ist ein Web-Client):

```
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
```

Eingangspaket von LAN- oder WAN-Seite zum Container

In diesem speziellen Szenario stammt das verfolgte Paket aus dem LAN. Aus Sicht der Umleitung gibt es relevante Unterschiede, wenn der Datenfluss vom LAN oder WAN ausgeht.

Der Client versucht über HTTPS auf www.cisco.com zuzugreifen.

```
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>
```

Der Datenverkehr, der mit der Bedingung übereinstimmt, wird auf der GigabitEthernet2-Schnittstelle nachverfolgt.

```
  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>
```

Auf dem Ausgangs-Feature-Invocation-Array (FIA) der Ausgangs-Schnittstelle beschloss die UTD FIA, dieses Paket in den Container umzuleiten.

```
  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>
```

Das Paket wird auf dem Standard-Tunnel Tunnel60001 platziert und über die VPG1-Schnittstelle weitergeleitet. In dieser Phase ist das ursprüngliche Paket GRE-gekapselt.

```
  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
```

Das Paket wird intern an den Container übertragen.

> **Anmerkung:** Weitere Informationen in diesem Abschnitt über Containerinternale werden nur zu Informationszwecken bereitgestellt. Der Zugriff auf den UTD-Container ist nicht über die normale CLI-Schnittstelle möglich.

Wenn der Datenverkehr tiefer im Router selbst geht, erreicht er eine interne VRF-Instanz an der Schnittstelle des Routingprozessors 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 ist eine mit dem IOSd-Prozess verbundene TIPC-Schnittstelle (Transport Inter Process Communication). Der OneP-Kanal wird darüber ausgeführt, um Konfigurationen und Benachrichtigungen zwischen dem IOSd-Container und dem UTD-Container hin und her zu übergeben.

Aus Ihrer Sicht wird "eth2 [Containerschnittstelle]" zu "VPG1 [192.0.2.1/192.168.2.2]" überbrückt. Dabei handelt es sich um die von vManage an IOS-XE und den Container gesendeten Adressen.

Wenn Sie **tcpdump** ausführen, wird der gekapselte GRE-Datenverkehr zum Container geleitet. Die GRE-Kapselung enthält einen 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.....
```

Eingangspaket vom Container zur LAN- oder WAN-Seite

Nach der Snort-Verarbeitung (unter der Annahme, dass der Datenverkehr nicht verworfen werden soll) wird er zurück in den QFP-Weiterleitungspfad geleitet.

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

Tunnel600001 ist die Ausgangsschnittstelle des Containers.

```
  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
```

Da der Datenverkehr bereits überprüft wurde, weiß der Router, dass es sich um eine erneute Injektion handelt.

```
  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
```

Der Datenverkehr wird NATed empfangen und geht zum Internet.

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

Integration von UTD-Flow-Protokollierung mit Packet-Trace

IOS-XE 17.5.1 hat die Integration von UTD-Flussprotokollierung mit Packet-Trace hinzugefügt, wobei die Pfadverfolgung-Ausgabe ein UTD-Urteil enthält. Bei Verdict kann es sich um Folgendes handeln, z. B.:

- das Paket, das UTD blockiert/warnt für Snort
- allow/drop für URLF
- Blockierung/Freigabe für AMP

Bei Paketen ohne UTD-Verdict-Informationen werden keine Flow-Protokollierungsinformationen protokolliert. Beachten Sie außerdem, dass keine Protokollierung von IPS/IDS-Pass/allow-Verdict aufgrund potenzieller negativer Auswirkungen auf die Leistung erfolgt.

Um die Integration der Flow-Protokollierung zu aktivieren, verwenden Sie die CLI-Add-On-Vorlage mit:

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

Beispielausgabe für verschiedene Urteile:

Timeout für URL-Suche:

```
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)
```

URL-Reputation und Urteilsberechtigung:

```
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
```

URLF-Reputation und Urteilsblock:

```
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
```

Voraussetzung:

Überprüfen der Kompatibilität der UTD-Version mit 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)
```

Wenn "UNSUPPORTED" angezeigt wird, ist das Container-Upgrade als erster Schritt vor Beginn der Fehlerbehebung erforderlich.

Überprüfen Sie, ob der Container eine gültige Namenserver-Konfiguration enthält.

Bei einigen Sicherheitsdiensten wie AMP und URLF muss der UTD-Container in der Lage sein, Namen für Cloud-Service-Provider aufzulösen. Daher muss der UTD-Container gültige Namenserver-Konfigurationen aufweisen. Dies kann überprüft werden, indem die Datei resolv.conf für den Container unter der System-Shell überprüft wird:

```
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
```

Problem 1

Die Unified Thread Defense-Funktion muss für jedes Design zusammen mit dem Direct Internet Access-Anwendungsfall (DIA) konfiguriert werden. Der Container versucht, **api.bcti.brightcloud.com** aufzulösen, um URL-Reputationen und -Kategorien abzufragen. In diesem Beispiel werden keine der inspizierten URLs blockiert, selbst wenn die richtige Konfiguration angewendet wird

Fehlerbehebung

Überprüfen Sie immer die Containerprotokolldatei.

```
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
```

Dadurch wird die Protokolldatei im Flash selbst kopiert.

Die Anzeige des Protokolls kann mit dem folgenden Befehl erfolgen:

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

Das Anzeigen des Protokolls zeigt Folgendes auf:

```
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
```

Standardmäßig stellt vManage einen Container bereit, der OpenDNS-Server verwendet [208.67.222.222 und 208.67.220.220].

Ursache

Der DNS-Datenverkehr (Domain Name System) zur Auflösung von **api.bcti.heltcloud.com** wird irgendwo im Pfad zwischen dem Container und den übergeordneten DNS-Servern verworfen. Stellen Sie sicher, dass beide DNS-Adressen erreichbar sind.

Problem 2

In einem Szenario, in dem Websites der Kategorie Computer und Internetinfo blockiert werden sollen, wird die HTTP-Anforderung an www.cisco.com korrekt gelöscht, während HTTPS-Anfragen nicht zulässig sind.

Fehlerbehebung

Wie bereits erläutert, wird der Datenverkehr an den Container geleitet. Wenn dieser Fluss in den GRE-Header eingekapselt wird, wird Software sowie ein VPATH-Header hinzugefügt. Mithilfe dieses Headers kann das System eine Debugbedingung an den Container selbst übergeben. Das bedeutet, UTD-Container sind gut zu warten.

In diesem Szenario lautet die Client-IP-Adresse 192.168,16.254. Beheben wir nun die Fehlerbehebung für den von meinem Client stammenden Datenverkehr durch den Container selbst.

```
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
```

Dieser Befehlssatz weist IOS-XE an, Datenverkehr von oder nach 192.168.16.254 zu markieren. Dadurch kann Debug-me-Flag über den VPATH-Header an den Container übergeben werden.

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 only this specific flow, while other are processing normal.

In dieser Phase können Sie den Benutzer auffordern, den Datenverkehr vom Client zu www.cisco.com auszulösen.

Im nächsten Schritt werden die Protokolle abgerufen:

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

Bei HTTP-Datenverkehr erkennt der HTTP-Präprozessor die URL in der Get-Anforderung.

```
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
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Bei HTTPS-Datenverkehr wurde der Ziel-DNS vom HTTPS-Präprozessor aus dem Hello extrahiert

```
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 sehen Sie nicht, dass die Blockierungsseite ausgelöst wird, da die Software die Ergebnisse der Webroot-Abfrage nicht meldet.

Ursache

CSCvo 77664 "UTD-URL-Filterung für Kategoriesuche schlägt fehl, wenn Webroot-Suche fehlschlägt" bezieht sich auf den Datenverkehr, der verloren geht, wenn die Software noch keine Antwort auf unsere URL-Urteilsanforderung hat.

Problem 3

In diesem Szenario werden gelegentlich Webbrowsersitzungen, die durch die URL-Filterung [aufgrund ihrer Klassifizierung] zugelassen werden sollen, verworfen. Beispielsweise ist der Zugriff auf www.google.com zufällig nicht möglich, auch wenn die Kategorie "Web-Suchmaschine" zugelassen ist.

Fehlerbehebung

Schritt 1: Sammeln allgemeiner Statistiken

Hinweis: Diese Befehlsausgabe wird alle 5 Minuten zurückgesetzt.

```
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" - steht für das HTTP GET- oder das HTTPS-Client/Server-Zertifikat [, wo SNI/DN für die Suche extrahiert werden kann. Verspätete Anfrage wird weitergeleitet.
- "sehr späte Anfragen" bedeutet, dass eine Art von Session-Drop-Zähler, bei dem weitere Pakete im Flow verworfen werden, bis der Router ein URL-Urteil von Brightcloud erhält. Mit anderen Worten, alles, was nach dem ersten HTTP GET oder dem Rest des SSL-Datenflusses passiert, wird verworfen, bis ein Urteil empfangen wird.
- "extrem späte Anfragen" - wenn die Sitzungsabfrage zu Brightcloud zurückgesetzt wurde, ohne ein Urteil abzugeben. Die Sitzung wird nach 60 Sekunden für Version < 17.2.1 beendet. Ab dem 17.2.1 wird die Abfrage in Brightcloud nach 2 Sekunden beendet. [über CSCvr98723 UTD: Timeout-URL-Anfragen nach zwei Sekunden]

In diesem Szenario sehen wir globale Zähler, die auf eine ungesunde Situation hinweisen.

Schritt 2: Überprüfen der Protokolldatei der Anwendung

Die Unified Thread Detection-Software zeichnet Ereignisse in der Anwendungsprotokolldatei auf.

```
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
```

Dadurch wird die Protokolldatei der Containeranwendung extrahiert und im Flash selbst gespeichert.

Die Anzeige des Protokolls kann mit dem folgenden Befehl erfolgen:

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

> **Hinweis:** In der IOS-XE-Softwareversion 20.6.1 und höher muss das UTD-Anwendungsprotokoll nicht mehr manuell verschoben werden. Diese Protokolle können jetzt mit dem Standardbefehl **show logging process vman module** angezeigt werden.

Das Anzeigen des Protokolls zeigt Folgendes auf:

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

- "FEHLER: Kann nicht an host api.bcti.brightcloud.com senden" - bedeutet, dass die Abfrage-Sitzung an Brightcloud abgelaufen ist [60 Sekunden < 17.2.1 / 2 Sekunden >= 17.2.1]. Dies ist ein Zeichen für eine schlechte Verbindung zu Brightcloud.
  Um das Problem zu demonstrieren, würde die Verwendung von EPC [Embedded Packet Capture] es ermöglichen, das Verbindungsproblem zu visualisieren.
- "SPP-URL-FILTERING txn_id misse match verdict" - Dieser Fehlerzustand erfordert etwas mehr Erklärung. Brightcloud-Abfrage wird über einen POST durchgeführt, bei dem der Router eine Abfrage-ID generiert.

Problem 4

In diesem Szenario ist IPS die einzige Sicherheitsfunktion, die in UTD aktiviert ist, und der Kunde hat Probleme mit der Druckerkommunikation, einer TCP-Anwendung.

Fehlerbehebung

Um dieses Datenath-Problem zu beheben, nehmen Sie zunächst die Paketerfassung vom TCP-Host, der das Problem hat. Die Erfassung zeigt einen erfolgreichen 3-Wege-TCP-Handshake, aber nachfolgende Datenpakete mit TCP-Daten scheinen vom cEdge-Router verworfen worden zu sein. Aktivieren Sie anschließend Packet-Trace, der Folgendes zeigte:

```
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)
```

Die oben angegebene Ausgabe mit den Paketnummern 8 und 9 wurde an die UTD-Engine weitergeleitet, aber sie wurden nicht erneut in den Weiterleitungspfad eingespeist. Durch das Überprüfen der UTD-Engine-Protokollierungsereignisse werden auch keine Verwerfungen bei der Snort-Signatur festgestellt. Überprüfen Sie anschließend die internen UTD-Statistiken, die aufgrund des TCP-Normalisierungsprogramms einige Paketverluste aufdecken:

```
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
```

Ursache

Die Ursache des Problems liegt in einem fehlerhaften TCP-Stack auf den Druckern. Wenn die Timestamp-Option während des TCP-Handshake in drei Richtungen ausgehandelt wird, gibt RFC7323 in jedem Nicht-<RST>-Paket eine TCP-MUSS-Option zum Senden von TSopt an. Eine genauere Untersuchung der Paketerfassung zeigt, dass die TCP-Datenpakete, die verworfen werden, nicht aktiviert sind. Bei der IOS-XE UTD-Implementierung ist Snort TCP Normalizer mit der Blockoption unabhängig von IPS oder IDS aktiviert.

Referenzen

- Leitfaden zur Sicherheitskonfiguration: Einheitlicher Schutz vor Bedrohungen