UTDおよびURLフィルタリングによるデータパス処理のトラブルシューティング

内容

概要

このドキュメントでは、IOS® XE WAN EdgesルータでSnortおよびUniform Resource Locator(URL)フィルタリングとも呼ばれるUnified Threat Defense(UTD)のトラブルシューティング方法について説明します。

背景説明

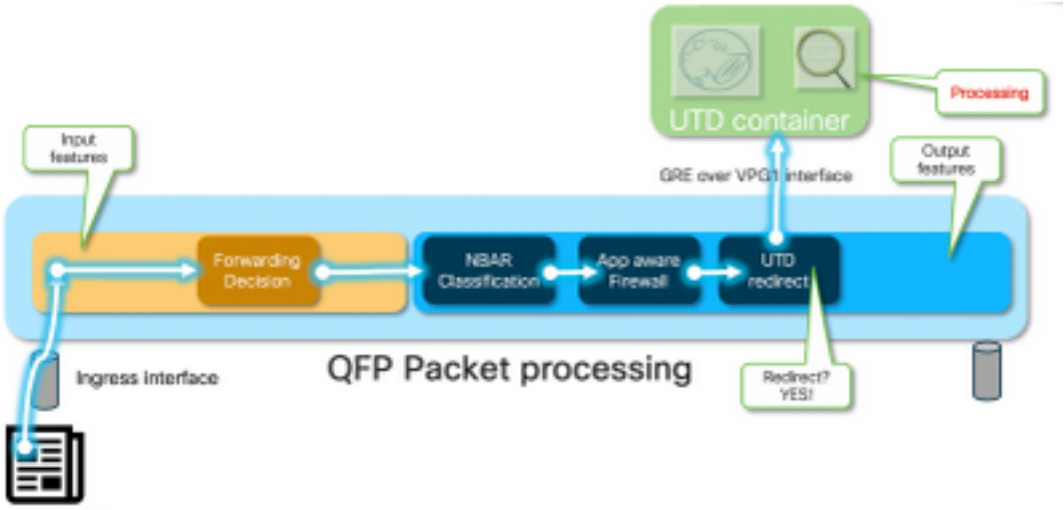
Snortは、世界で最も広く導入されている侵入防御システム(IPS)です。2013年よりは、Snortソフトウェアの商用バージョンを作成した会社であるSourcefireはシスコによって買収されます。16.10.1 IOS® XE SD-WANソフトウェア以降、UTD/URF-FilteringコンテナがCisco SD-WANソリューションに追加されました。

app-navフレームワークを使用して、コンテナがIOS® XEルータに登録されます。このプロセスの説明は、このドキュメントの範囲外です。

データパスの概要

データパスの概要は次のようになります。

LAN/WANからコンテナへ



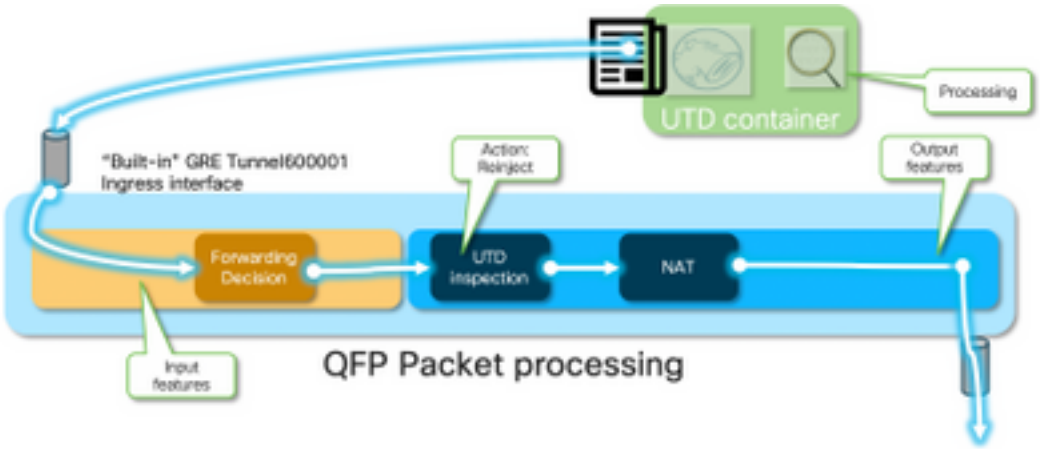
トラフィックはLAN側から送信されます。IOS® XEは、コンテナが正常な状態であることを認識しているため、トラフィックをUTDコンテナに転送します。この変換では、出力インターフェイスとしてVirtualPortGroup1インターフェイスが使用され、Generic Routing Encapsulation(GRE)トンネル内のパケットがカプセル化されます。

ルータは原因：64（サービスエンジンパケット）を使用して「PUNT」アクションを実行し、ルートプロセッサ(RP)にトラフィックを送信します。パントヘッダーが追加され、コンテナ「[internal0/0/svc_eng:0]」への内部出力インターフェイスを使用してパケットがコンテナに送信されます

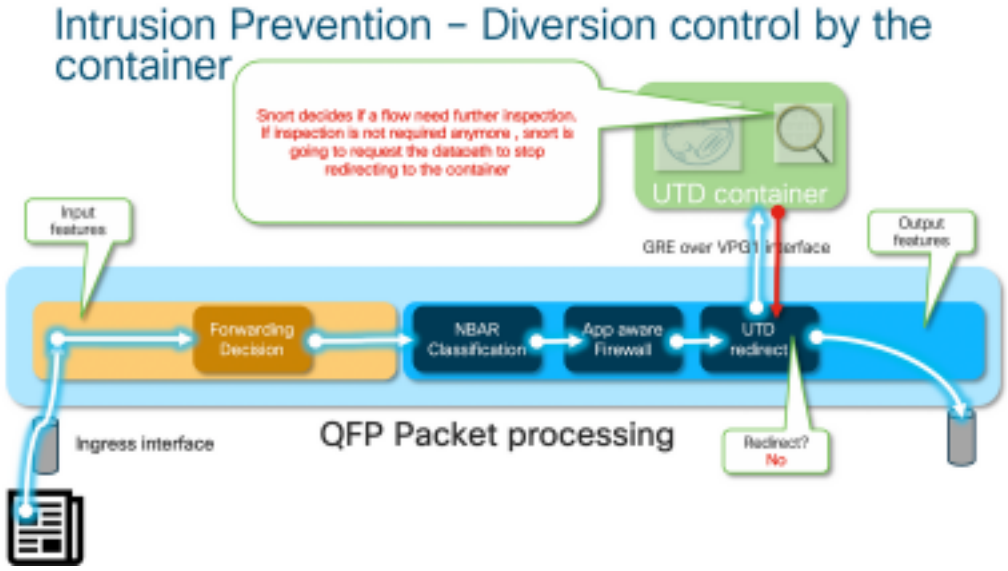
この段階で、Snortはプリプロセッサとルールセットを活用します。パケットは、処理結果に基づいてドロップまたは転送できます。

コンテナからLAN/WANへ

トラフィックがドロップされないものと仮定すると、パケットはUTD処理の後にルータに戻されます。これはQuantum Flow Processor(QFP)でTunnel6000001から送信されたものとして表示されます。その後、ルータによって処理され、WANインターフェイスにルーティングされる必要があります。



コンテナは、IOS® XEデータパスでのUTDインスペクションの結果として発生する転用を制御します。



たとえば、HTTPSフローの場合、プリプロセッサはTLSネゴシエーションを使用してサーバのHello/クライアントのHelloパケットを確認します。その後、フローはリダイレクトされません。これは、TLS暗号化トラフィックを検査する際の値がほとんどないためです。

データパスディープダイブ

パケットトレーサの観点から、これらの一連のアクションが表示されます(192.168.16.254はWebクライアントです)。

```
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
```

LANまたはWAN側からコンテナへの入力パケット

この特定のシナリオでは、トレースされたパケットはLANから送信されます。リダイレクションの観点からは、フローがLANまたはWANから来る場合は、関連する違いがあります。

クライアントはHTTPSでwww.cisco.comにアクセスしようと試みます

```
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>
```

条件に一致するトラフィックは、インターフェイスGigabitEthernet2の入力からトレースされます。

```
  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>
```

出力インターフェイスの出力Feature Invocation Array(FIA)で、UTD FIAはこのパケットをコンテナに転送することを決定しました。

```
  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>
```

パケットはデフォルトのトンネルTunnel600001に配置され、VPG1インターフェイス経由でルーティングされます。この段階では、元のパケットはGREカプセル化されます。

```
  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
```

パケットは内部でコンテナに送信されます。

> **注**：この項のコンテナ内部の詳細は、情報の目的でのみ提供されます。UTDコンテナには、通常のCLIインターフェイスからアクセスできません。

ルータ自体の奥深くに入ると、トラフィックはルートプロセッサインターフェイスeth2の内部VRFに到達します。

```
[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は、IOSdプロセスに接続されたTransport Inter Process Communication(TIPC)インターフェイスです。OnePチャネルは、IOSdコンテナとUTDコンテナの間で設定と通知を送受信するために、その上で動作します。

ご注意ください。「eth2 [container interface]」は、「VPG1 [192.0.2.1/192.168.2.2]」にブリッジされます。これは、vManageによってIOS-XEとコンテナにプッシュされるアドレスです。

tcpdumpを実行すると、コンテナに向かうGREカプセル化トラフィックを確認できます。GREカプセル化にはVPATHヘッダーが含まれます。

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

コンテナからLANまたはWAN側への入力パケット

Snort処理（トラフィックがドロップされないものと仮定）の後、QFP転送パスに再注入されます。

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

Tunnel60001は、コンテナからの出力インターフェイスです。

```
  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
```

トラフィックはすでに検査されているため、ルータはこれを再注入であると認識しています。

```
  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
```

トラフィックはNAT処理され、インターネットに向けて送信されます。

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

パケットトレースとのUTDフローロギング統合

IOS-XE 17.5.1では、パケットトレースとのUTDフローロギング統合が追加されました。この場合、パストレース出力にはUTD判定が含まれます。判定は、次のいずれかになります。

- UTDがSnortのブロック/アラートを決定したパケット
- URLFの許可/ドロップ
- AMPのブロック/許可

UTD判定情報を持たないパケットの場合、フローロギング情報は記録されません。また、パフォーマンスに悪影響を及ぼす可能性があるため、IPS/IDS pass/allow判定のロギングはありません。

フローロギング統合を有効にするには、次のCLIアドオンテンプレートを使用します。

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

さまざまな用語の出力例：

URL検索タイムアウト：

```
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)
```

URLFレピュテーションと判定の許可：

```
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レピュテーションと判定ブロック：

```
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
```

前提条件：

UTDバージョンがIOS XEと互換性があるかどうかを確認する

```
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)
```

「UNSUPPORTED」と表示されている場合は、トラブルシューティングを開始する前に、最初のステップとしてコンテナのアップグレードが必要です。

コンテナ内の有効なネームサーバ設定を確認します

AMPやURLFなどのセキュリティサービスの一部では、UTDコンテナがクラウドサービスプロバイダーの名前を解決できる必要があるため、UTDコンテナには有効なネームサーバ設定が必要です。これは、システムシェルのコンテナのresolv.confファイルを確認することで確認できます。

```
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
```

問題 1

設計に従って、Unified Thread Defenseは、Direct Internet Accessのユースケース(DIA)と一緒に

設定する必要があります。 コンテナは、URLのレピュテーションとカテゴリを照会するために api.bcti.brightcloud.comを解決しようとします。この例では、適切な設定が適用されても、検査されたURLはブロックされません

トラブルシュート

コンテナログファイルを常に確認してください。

```
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
```

これにより、フラッシュ自体にログファイルがコピーされます。

ログを表示するには、次のコマンドを使用します。

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

ログの表示：

```
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
```

デフォルトでは、vManageはOpenDNSサーバ[208.67.222.222および208.67.220.220]を使用するコンテナをプロビジョニングします

根本原因

api.bcti.brightcloud.comを解決するドメインネームシステム(DNS)のトラフィックは、コンテナと包括的なDNSサーバ間のパスのどこかにドロップされます。必ず両方のDNSが到達可能であることを確認してください。

問題 2

Computer and Internet Infoカテゴリのwebサイトがブロックされる場合は、HTTPS要求がブロックされていない間、www.cisco.comへのHTTP要求が正しくドロップされます。

トラブルシュート

前に説明したように、トラフィックはコンテナにパントされます。このフローがGREヘッダーにカプセル化されると、ソフトウェアはVPATHヘッダーと同様に追加します。このヘッダーを使用すると、システムはデバッグ条件をコンテナ自体に渡すことができます。これは、UTDコンテナがサービス可能であることを意味します。

このシナリオでは、クライアントのIPアドレスは192.168,16.254です。クライアントからのトラフィックに対してコンテナ自体によるSnort処理をトラブルシューティングしてみましょう。

```
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
```

この一連のコマンドは、192.168.16.254との間でトラフィックをマークするようにIOS-XEに指示します。これにより、debug-meフラグがVPATHヘッダーを介してコンテナに渡されます

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のデバッグでは、特定のフローだけがデバッグされ、他のフローは正常に処理されます。

この段階で、クライアントからwww.cisco.comへのトラフィックをトリガーするようにユーザに依頼できます。

次の手順は、ログを取得することです。

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

HTTPトラフィックの場合、Snort HTTPプリプロセッサはget要求でURL Inを検出します。

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

httpsトラフィックの場合、宛先DNSはHTTPSプリプロセッサによってサーバhelloから抽出されています

```
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
```

ソフトウェアがWebrootクエリの結果をレポートしないため、ブロッキングページがトリガーされないことがわかります。

根本原因

CSCvo77664 「カテゴリ検索のUTD URLフィルタリングがwebrootルックアップ失敗で失敗しました」は、ソフトウェアがURL判定要求に応答していないときにトラフィックがリークされている問題です。

問題3

このシナリオでは、URLフィルタリング（分類のため）で許可されるWebブラウジングセッションが断続的にドロップされます。例えば、www.google.comへのアクセスは、カテゴリ「Web検索エンジン」が許可されている場合でもランダムに行うことはできません。

トラブルシュート

ステップ1:一般統計の収集

注：このコマンド出力は5分ごとにリセットされます

```
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」:HTTP GETまたはHTTPSクライアント/サーバ証明書[ここでSNI/DNを検索に抽出できます。遅延要求が転送されます。
- 「非常に遅い要求」：ルータがBrightcloudからURL判定を受信するまで、フロー内のパケッ

トがドロップされる何らかのセッションドロップカウンタを意味します。つまり、最初のHTTP GETの後、またはSSLフローの残りはすべて、判定が受信されるまで廃棄されます。
- 「非常に遅いリクエスト」：判定を行わずにBrightcloudへのセッションクエリがリセットされた場合。バージョン< 17.2.1のセッションは60秒後にタイムアウトします。17.2.1以降では、Brightcloudへのクエリセッションは2秒後にタイムアウトします。[CSCvr98723経由 UTD:2秒後にURL要求をタイムアウト]

このシナリオでは、異常な状況を示すグローバルカウンタが表示されます。

ステップ2:アプリケーションログファイルを見る

Unified Thread Detectionソフトウェアは、アプリケーションログファイルにイベントを記録します。

```
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
```

コンテナアプリケーションのログファイルを抽出し、フラッシュに保存します。

ログを表示するには、次のコマンドを使用します。

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

> **注：**IOS-XEソフトウェアバージョン20.6.1以降では、UTDアプリケーションログを手動で移動する必要はなくなりました。これらのログは、標準コマンド**show logging process vman module utdを使用して表示できます**

ログの表示：

```
.....
2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 245 , utmdata
txn_id 0 2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 248 ,
utmdata txn_id 0 2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id
249 , utmdata txn_id 0 2020-04-14 17:47:57.504:(#1):SPP-URL-FILTERING txn_id miss match verdict
txn_id 250 , utmdata txn_id 0 2020-04-14 17:47:57.660:(#1):SPP-URL-FILTERING txn_id miss match
verdict txn_id 251 , utmdata txn_id 0 2020-04-14 17:47:57.660:(#1):SPP-URL-FILTERING txn_id miss
match verdict txn_id 254 , utmdata txn_id 0 2020-04-14 17:47:57.660:(#1):SPP-URL-FILTERING
txn_id miss match verdict txn_id 255 , utmdata txn_id 0 2020-04-14 17:48:05.725:(#1):SPP-URL-
FILTERING txn_id miss match verdict txn_id 192 , utmdata txn_id 0 2020-04-14
17:48:37.629:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 208 , utmdata txn_id 0
2020-04-14 17:49:55.421:(#1):SPP-URL-FILTERING txn_id miss match verdict txn_id 211 , utmdata
txn_id 0 2020-04-14 17:51:40 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:52:21 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:52:21 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:53:56 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:54:28 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:54:29 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out 2020-04-14 17:54:37 ERROR: Cannot send to host api.bcti.brightcloud.com: Connection
timed out
....
```

- 「Error:Cannot send to host api.bcti.brightcloud.com" - Brightcloudへのクエリセッションがタイムアウトしたことを意味します[60秒< 17.2.1 / 2秒>= 17.2.1]。これは、Brightcloudへの接続不良の兆候です。

この問題を実証するために、EPC [Embedded Packet Capture]を使用すると、接続の問題を可視化できます。
- 「SPP-URL-FILTERING txn_id miss match verdict」 – このエラー状態には少し詳しい説明が必要です。Brightcloudクエリは、ルータによってクエリIDが生成されるPOSTを介して実行されます

問題 4

このシナリオでは、UTDで有効になっている唯一のセキュリティ機能はIPSであり、TCPアプリケーションであるプリンタ通信に関する問題が発生しています。

トラブルシュート

このデータパスの問題をトラブルシューティングするには、まず問題のあるTCPホストからパケットキャプチャを取得します。このキャプチャは、TCP 3ウェイハンドシェイクが成功したことを示していますが、TCPデータを含む後続のデータパケットがcEdgeルータによってドロップされたように見えます。次に、パケットトレースを有効にします。次の例を示します。

```
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)
```

上記の出力は、パケット番号8と9がUTDエンジンに転送されましたが、フォワーディングパスに再注入されなかったことを示しています。UTDエンジンの口ギングイベントをチェックしても、Snortシグニチャのドロップは明らかになりません。次に、UTDの内部統計情報を確認します。これにより、TCPノーマライザによるパケットドロップが明らかになります。

```
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
```

根本原因

この問題の根本原因は、プリンタのTCPスタックの誤動作にあります。TCP 3ウェイハンドシェイク中にTimestampオプションがネゴシエートされると、RFC7323はTCPが非<RST>パケットご

とにTSoptオプションを送信する必要があると規定します。パケットキャプチャを詳しく調べると、ドロップされるTCPデータパケットで次のオプションが有効になっていないことが分かります。IOS-XE UTDの実装では、IPSまたはIDSに関係なく、ブロックオプションを使用したSnort TCPノーマライザが有効になります。

参考資料

- セキュリティの設定ガイド：統合脅威防御