# Contents

# Introduction

This document describes the configuration required for passing AVC traffic through an IPSEC tunnel to the collector. By default, AVC information cannot be exported across an IPSEC tunnel to the collector

# Prerequisites

Cisco recommends that you have the basic knowledge of these topics:

- Application Visibility and Control (AVC)
- Easy Performance Monitor (EzPM)

# Background Information

The Cisco AVC feature is used to recognize, analyze and control over multiple applications. With application awareness built into the network infrastructure, plus visibility into the performance of applications running on the network, AVC enables per-application policy for granular control of application bandwidth use, resulting in a better end user experience. Here you can find more details about this technology.

EzPM is a faster and easier way to configure the traditional performance monitoring configuration. Currently EzPM does not provides the full flexibility of the traditional performance monitor configuration model. Here you can find more details about EzPM.

# Limitation

Currently AVC does not support the number of pass-through tunneling protocols, details can be found here.

Internet Protocol Security (IPSec) is one of the unsupported pass-through tunneling protocols for AVC and this document addresses the possible workaround for this limitation.

# Configure

This section describes the complete configuration used to simulate the given limitation.

## Network Diagram

In this network diagram all the routers have reachability to each other using the static routes. R1 is configured with the EzPM configuration and has one IPSec tunnel established with R2 router. R3 is working as an exporter here, which could be Cisco Prime or any other kind of exporter which is capable of collecting the performance data.

AVC traffic is generated by R1 and it is sent to the exporter via R2. R1 sends the AVC traffic to R2 over an IPSec tunnel interface.

## Initial configuration

This section describes the initial configuration for R1 through R3.

**R1**

```
!
interface Loopback0
ip address 1.1.1.1 255.255.255.255
!

interface GigabitEthernet0/1

 ip address 172.16.1.1 255.255.255.0

 duplex auto

 speed auto

!

ip route 0.0.0.0 0.0.0.0 172.16.1.2

!
```

**R2**

```
!
interface GigabitEthernet0/0/0
 ip address 172.16.2.2 255.255.255.0
 negotiation auto
 !
interface GigabitEthernet0/0/1
 ip address 172.16.1.2 255.255.255.0
 negotiation auto
!
```

**R3**

```
!
interface GigabitEthernet0/0
 ip address 172.16.2.1 255.255.255.0
 duplex auto
 speed auto
!
ip route 0.0.0.0 0.0.0.0 172.16.2.2
!
```

## IPSec configuration

This section describes the IPSec configuration for R1 and R2 router.

**R1**

```
!
ip access-list extended IPSec_Match
 permit ip any host 172.16.2.1
!
```

```
crypto isakmp policy 1
 encr aes 256
 hash md5
 authentication pre-share
 group 2
crypto isakmp key cisco123 address 172.16.1.2
!
!
crypto ipsec transform-set set2 esp-aes 256 esp-sha-hmac
 mode tunnel
!
!
crypto map VPN 10 ipsec-isakmp
 set peer 172.16.1.2
 set transform-set set2
 match address IPSec_Match
!
interface GigabitEthernet0/1
 ip address 172.16.1.1 255.255.255.0
 duplex auto
 speed auto
 crypto map VPN
!
```

**R2**

```
!
ip access-list extended IPSec_Match
```

```
 permit ip host 172.16.2.1 any
!
crypto isakmp policy 1
 encr aes 256
 hash md5
 authentication pre-share
 group 2
crypto isakmp key cisco123 address 172.16.1.1
!
!
crypto ipsec transform-set set2 esp-aes 256 esp-sha-hmac
 mode tunnel
!
!
crypto map VPN 10 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set set2
 match address IPSec_Match
 reverse-route
!
interface GigabitEthernet0/0/1
 ip address 172.16.1.2 255.255.255.0
 negotiation auto
 cdp enable
 crypto map VPN
!
```

To verify whether the IPSec config is working as expected or not, check the output for **show**

**crypto isakmp sa**

```
R1#show crypto isakmp sa

IPv4 Crypto ISAKMP SA

dst             src             state          conn-id status

IPv6 Crypto ISAKMP SA
```

In order to bring the security associations up, ping the exporter (R3, 172.16.2.1) from R1.

```
R1#ping 172.16.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

R1#
```

Now, the router will have an active security association, which confirms that the traffic being originated from R1 and destined to the exporter is ESP encapsulated.

```
R1#show crypto isakmp sa

IPv4 Crypto ISAKMP SA

dst             src             state          conn-id status

172.16.1.2      172.16.1.1      QM_IDLE           1002 ACTIVE

IPv6 Crypto ISAKMP SA
```

## EzPM configuration

This section describes the EzPM configuration for R1 router.

**R1**

```
!

class-map match-all perf-mon-acl

 description PrimeAM generated entity - do not modify or use this entity

 match protocol ip

!
```

```
performance monitor context Performance-Monitor profile application-experience

 exporter destination 172.16.2.1 source GigabitEthernet0/1 transport udp port 9991

 traffic-monitor application-traffic-stats

 traffic-monitor conversation-traffic-stats ipv4

 traffic-monitor application-response-time ipv4

 traffic-monitor media ipv4 ingress

 traffic-monitor media ipv4 egress

 traffic-monitor url ipv4 class-replace perf-mon-acl

!
```

Apply the EzPM profile on the interface which needs to be monitored; here we are monitoring the loopback 0 interface.

**R1**

```
!

interface Loopback0

 ip address 1.1.1.1 255.255.255.255

 performance monitor context Performance-Monitor

!
```

## Workaround

With the above configuration in place, take the output for **show performance monitor context***context-name***exporter**.

Check for the status of **Output Features** option, by default it should be in **Not Used** state, which is an expected behavior and that is why the AVC traffic is not being encapsulated or encrypted here.

In order to let the AVC traffic pass through the IPsec tunnel interface, **Output Features** option shall be in used state. And to do that, it has to be enabled explicitly in flow exporter profile. Below is the detailed step by step procedure to enable this option.

**Step-1**

Take the complete output for **show performance monitor context** *context-name* **configuration** command and save it in notepad. Below is the snip for this output,

```
R1#show performance monitor context Performance-Monitor configuration
```

```
!=======================================================================
========
!            Equivalent Configuration of Context Performance-
Monitor           !
!=======================================================================
========
!Exporters
!==========
!
flow exporter Performance-Monitor-1
 description performance monitor context Performance-Monitor exporter
 destination 172.16.2.1
 source GigabitEthernet0/1
 transport udp 9991
 export-protocol ipfix
 template data timeout 300
 option interface-table timeout 300
 option vrf-table timeout 300
 option c3pl-class-table timeout 300
 option c3pl-policy-table timeout 300
 option sampler-table timeout 300
 option application-table timeout 300
 option application-attributes timeout 300
 option sub-application-table timeout 300
-----------------------snip------------------------
```

**Step-2**

Add the **output-features** option explicitly under the flow exporter profile. After adding the output-features option the flow exporter profile shall look like this,

flow exporter Performance-Monitor-1

```
 description performance monitor context Performance-Monitor exporter
 destination 172.16.2.1
 source GigabitEthernet0/1
 transport udp 9991
 export-protocol ipfix
 template data timeout 300
output-features
 option interface-table timeout 300
 option vrf-table timeout 300
 option c3pl-class-table timeout 300
 option c3pl-policy-table timeout 300
 option sampler-table timeout 300
 option application-table timeout 300
 option application-attributes timeout 300
 option sub-application-table timeout 300
```

Leave the rest of the output as it is, DO NOT alter anything else in the output.

**Step-3**

Now, remove the EzPM profile from the Interface and from the router as well.

```
!
Interface loopback 0
no performance monitor context Performance-Monitor
exit
!
!
no performance monitor context Performance-Monitor profile application-experience
!
```

**Step-4**

Apply the modified config on the R1 router. Make sure that not a single command is missed out, since it may cause any unexpected behavior.

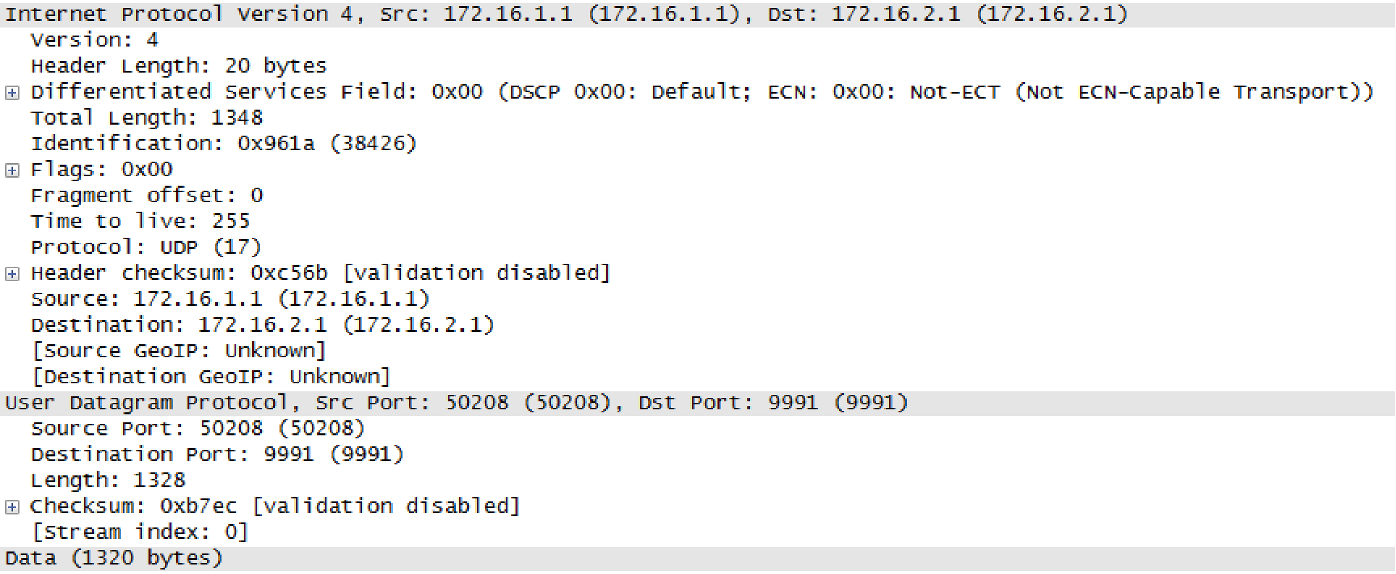
# Verify

This section describes the verification method used in this document to check and how this workaround has helped to overcome the limitation for AVC packets mentioned here.

Before applying the workaround, packets received by the IPSec peer router (R2) will be dropped. Below message will be generated as well:

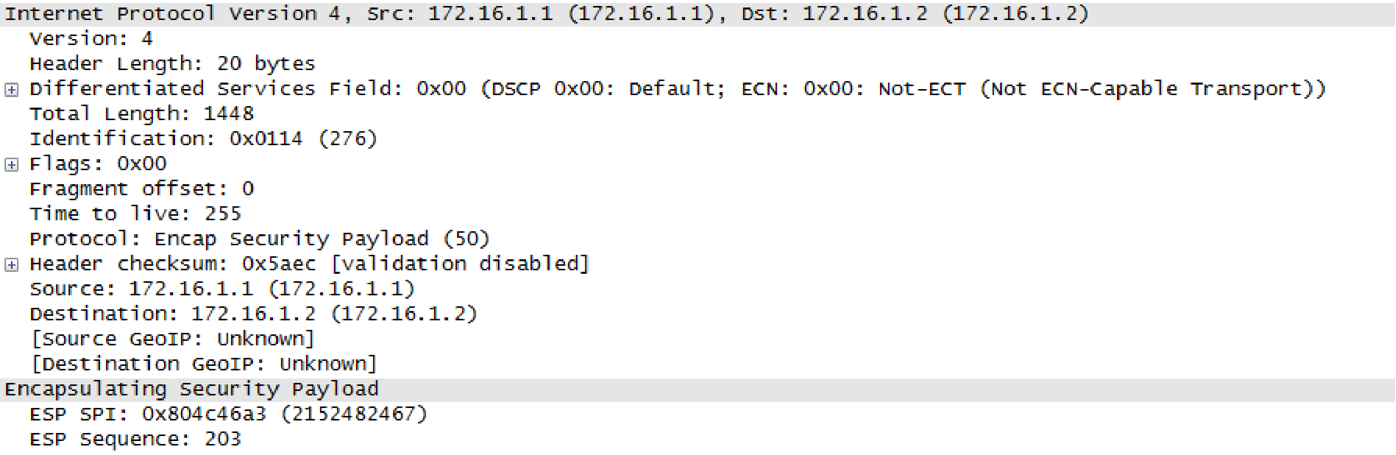
```
%IPSEC-3-RECVD_PKT_NOT_IPSEC: Rec'd packet not an IPSEC packet,
dest_addr= 172.16.2.1, src_addr= 172.16.1.1, prot= 17
```

Here R2 is expecting the ESP encapsulated packets which are destined for 172.16.2.1, but the received packets are plain UDP packets (prot=17) and it is an expected behavior to drop these packets. Below packet capture shows that the packet received at R2 is a plain UDP packet instead of ESP encapsulated, which is a default behavior for AVC.



```
Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.2.1 (172.16.2.1)
   Version: 4
   Header Length: 20 bytes
⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
   Total Length: 1348
   Identification: 0x961a (38426)
⊞ Flags: 0x00
   Fragment offset: 0
   Time to live: 255
   Protocol: UDP (17)
⊞ Header checksum: 0xc56b [validation disabled]
   Source: 172.16.1.1 (172.16.1.1)
   Destination: 172.16.2.1 (172.16.2.1)
   [Source GeoIP: Unknown]
   [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 50208 (50208), Dst Port: 9991 (9991)
   Source Port: 50208 (50208)
   Destination Port: 9991 (9991)
   Length: 1328
⊞ Checksum: 0xb7ec [validation disabled]
   [Stream index: 0]
Data (1320 bytes)
```

After applying the workaround, it is clearly seen from the below packet capture that the AVC packets received at R2 are ESP encapsulated and no more error messages seen on the R2.



```
Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.1.2 (172.16.1.2)
   Version: 4
   Header Length: 20 bytes
⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
   Total Length: 1448
   Identification: 0x0114 (276)
⊞ Flags: 0x00
   Fragment offset: 0
   Time to live: 255
   Protocol: Encap Security Payload (50)
⊞ Header checksum: 0x5aec [validation disabled]
   Source: 172.16.1.1 (172.16.1.1)
   Destination: 172.16.1.2 (172.16.1.2)
   [Source GeoIP: Unknown]
   [Destination GeoIP: Unknown]
Encapsulating Security Payload
   ESP SPI: 0x804c46a3 (2152482467)
   ESP Sequence: 203
```

# Troubleshooting

Currently there is no specific troubleshooting information available for this configuration.