

Configuración de IPsec en switches Catalyst serie 9000X

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Terminology](#)

[Configurar](#)

[Diagrama de la red](#)

[Instalar licencia HSEC](#)

[Protección de Túnel SVTI](#)

[Verificación](#)

[Túnel IPsec](#)

[Plano de control IOSd](#)

[Plano de control PD](#)

[Troubleshoot](#)

[IOSd](#)

[Plano de control PD](#)

[Plano de datos PD](#)

[Dataplane Packet-tracer](#)

[Depuración de PD Dataplane](#)

[Información Relacionada](#)

Introducción

Este documento describe cómo verificar la función de seguridad de protocolo de Internet (IPsec) en los switches Catalyst 9300X.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- IPsec

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- C9300X
- C9400X
- Cisco IOS® XE 17.6.4 y versiones posteriores

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

A partir de Cisco IOS® XE 17.5.1, los switches Catalyst serie 9300-X admiten IPsec. IPsec proporciona altos niveles de seguridad a través del cifrado y la autenticación, así como la protección de los datos contra el acceso no autorizado. La implementación de IPsec en el C9300X proporciona túneles seguros entre dos pares mediante la configuración sVTI (interfaz de túnel virtual estática).

La compatibilidad con IPsec en los switches Catalyst serie 9400-X se introdujo en Cisco IOS® XE 17.10.1, mientras que la compatibilidad con Catalyst 9500-X está programada para 17.12.1.

Terminology

IOSd	daemon de IOS	Este es el demonio del IOS de Cisco que se ejecuta en el kernel de Linux. Se ejecuta como un proceso de software dentro del kernel. IOSd procesa los comandos CLI y los protocolos que crean el estado y la configuración.
PD	Dependiente de la plataforma	Datos y comandos específicos de la plataforma en la que se ejecutan
IPsec	Seguridad de protocolo de Internet	Conjunto de protocolos de red seguros que autentica y cifra paquetes de datos para proporcionar una comunicación cifrada segura entre dos equipos a través de una red de protocolo de Internet.
sVTI	Interfaz de Túnel Virtual Estático	Una interfaz virtual configurada estáticamente a la que puede aplicar funciones de seguridad
SA	Asociación de seguridad	Una SA es una relación entre dos o más entidades que describe cómo las entidades utilizan los servicios de seguridad para

		comunicarse de forma segura
FED	Controlador de motor de reenvío	El componente del switch responsable de la programación de hardware de UADP ASIC

Configurar

Diagrama de la red

A efectos de este ejemplo, Catalyst 9300X y ASR1001-X funcionan como pares IPsec con interfaces de túnel virtual IPsec.



Instalar licencia HSEC

Active la función IPsec en la plataforma Catalyst 9300X; se necesita una licencia HSEC (C9000-HSEC). Esto es diferente de otras plataformas de ruteo basadas en Cisco IOS XE que soportan IPsec, donde una licencia HSEC solo es necesaria para aumentar el rendimiento de cifrado permitido. En la plataforma Catalyst 9300X, el modo de túnel y la CLI de protección de túnel se bloquean si no se instala una licencia HSEC:

```
<#root>
```

```
C9300X(config)#
```

```
int tunnel1
```

```
C9300X(config-if)#
```

```
tunnel mode ipsec ipv4
```

```
%'tunnel mode' change not allowed
```

```
*Sep 19 20:54:41.068: %PLATFORM_IPSEC_HSEC-3-INVALID_HSEC: HSEC
```

license not present: IPSec mode configuration is rejected

Instale la licencia HSEC cuando el switch esté conectado a CSSM o CSLU mediante Smart Licensing:

<#root>

C9300X#

```
license smart authorization request add hseck9 local
```

*Oct 12 20:01:36.680: %SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS: A new licensing authorization code wa

Verifique que la licencia HSEC esté instalada correctamente:

<#root>

C9300X#

```
show license summ
```

Account Information:

Smart Account: Cisco Systems, TAC As of Oct 13 15:50:35 2022 UTC

Virtual Account: CORE TAC

License Usage:

License	Entitlement Tag	Count	Status
network-advantage	(C9300X-12Y Network Adv...)	1	IN USE
dna-advantage	(C9300X-12Y DNA Advantage)	1	IN USE
C9K HSEC	(Cat9K HSEC)	0	

NOT IN USE

Habilite IPsec como el modo de túnel en la interfaz de túnel:

<#root>

C9300X(config)#

```
int tunnel1
```

C9300X(config-if)#

```
tunnel mode ipsec ipv4
```

C9300X(config-if)#

```
end
```

Una vez que se habilita IPsec, la licencia HSEC se convierte EN USO

```
<#root>
```

```
C9300X#
```

```
show license summ
```

```
Account Information:
```

```
Smart Account: Cisco Systems, TAC As of Oct 13 15:50:35 2022 UTC
```

```
Virtual Account: CORE TAC
```

```
License Usage:
```

```
License Entitlement Tag Count Status
```

```
-----  
network-advantage (C9300X-12Y Network Adv...) 1 IN USE
```

```
dna-advantage (C9300X-12Y DNA Advantage) 1 IN USE
```

```
C9K HSEC (Cat9K HSEC) 1
```

```
IN USE
```

Protección de Túnel SVTI

La configuración IPsec en el C9300X utiliza la configuración IPsec IOS XE estándar de Cisco. Se trata de una configuración sencilla de SVTI que utiliza [IKEv2 Smart Defaults](#), donde utilizamos la política IKEv2 predeterminada, la propuesta IKEv2, la transformación IPsec y el perfil IPsec para IKEv2.

Configuración de C9300X

```
<#root>
```

```
ip routing
```

```
!
```

```
crypto ikev2 profile default
```

```
match identity remote address 192.0.2.2 255.255.255.255
```

```
authentication remote pre-share key cisco123
```

```
authentication local pre-share key cisco123
```

```
!
```


```
interface Tunnel1
```

```
ip address 192.168.1.1 255.255.255.252
```

```
tunnel source 198.51.100.1
```

```
tunnel mode ipsec ipv4
```

```
tunnel destination 192.0.2.2
tunnel protection ipsec profile default
```

 Nota: Dado que Catalyst 9300X es básicamente un switch de capa de acceso, el routing ip debe habilitarse explícitamente para que funcionen las funciones basadas en routing, como VTI.

Configuración de Peer

<#root>

```
crypto ikev2 profile default
```

```
match identity remote address 198.51.100.1 255.255.255.255
authentication remote pre-share key cisco123
authentication local pre-share key cisco123
```

```
!
```

```
interface Tunnel1
```

```
ip address 192.168.1.2 255.255.255.252
tunnel source 192.0.2.2
tunnel mode ipsec ipv4
tunnel destination 198.51.100.1
```

```
tunnel protection ipsec profile default
```

Para obtener una descripción más detallada de las distintas construcciones de configuración de IKEv2 e IPsec, consulte la [Guía de configuración de IPsec en C9300X](#).

Verificación

Túnel IPsec

La implementación de IPsec en la plataforma C9300X es diferente desde el punto de vista arquitectónico que en las plataformas de routing (ASR1000, ISR4000, Catalyst 8200/8300, etc.), donde el procesamiento de funciones de IPsec se implementa en el microcódigo QFP (procesador de flujo cuántico).

La arquitectura de reenvío de C9300X se basa en UADP ASIC, por lo que la mayor parte de la implementación de FIA de la función QFP no se aplica aquí.

Estas son algunas de las diferencias clave:

- `show crypto ipsec sa peer x.x.x.x platform` no muestra la información de programación de la plataforma desde el FMAN hasta el QFP.

- Packet-trace tampoco funciona (más información sobre esto a continuación).
- UADP ASIC no admite la clasificación de tráfico criptográfico, por lo que no se aplica show crypto ruleset platform

Plano de control IOSd

La verificación del plano de control IPsec es exactamente la misma que para las plataformas de routing, consulte . Para mostrar la SA IPsec instalada en IOSd:

```
<#root>
```

```
C9300X#
```

```
show crypto ipsec sa
```

```
interface: Tunnel1
```

```
  Crypto map tag: Tunnel1-head-0, local addr 198.51.100.1
```

```
protected vrf: (none)
```

```
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
```

```
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
```

```
current_peer 192.0.2.2 port 500
```

```
  PERMIT, flags={origin_is_acl,}
```

```
  #pkts encaps: 200, #pkts encrypt: 200, #pkts digest: 200
```

```
  #pkts decaps: 200, #pkts decrypt: 200, #pkts verify: 200
```

```
  #pkts compressed: 0, #pkts decompressed: 0
```

```
  #pkts not compressed: 0, #pkts compr.
```

```
failed: 0
```

```
  #pkts not decompressed: 0, #pkts decompress failed: 0
```

```
#send errors 0, #recv errors 0
```

```
local crypto endpt.: 198.51.100.1, remote crypto endpt.: 192.0.2.2
```

```
plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb TwentyFiveGigE1/0/1
```

```
current outbound spi: 0x42709657(1114674775)
```

```
PFS (Y/N): N, DH group: none
```

```
inbound esp sas:
```

```
  spi: 0x4FE26715(1340237589)
```

```
  transform: esp-aes esp-sha-hmac ,
```

```
  in use settings ={Tunnel, }
```

```
  conn id: 2098,
```

```
flow_id: CAT9K:98
```

```
, sibling_flags FFFFFFFF80000048, crypto map: Tunnel1-head-0
```

```
  sa timing: remaining key lifetime (k/sec): (26/1605)
```

```
  IV size: 16 bytes
```

```
  replay detection support: Y
```

```
  Status: ACTIVE(ACTIVE)
```

```
inbound ah sas:
```

```
inbound pcp sas:
```

outbound esp sas:

spi: 0x42709657(1114674775)
transform: esp-aes esp-sha-hmac ,
in use settings ={Tunnel, }
conn id: 2097,

flow_id: CAT9K:97

, sibling_flags FFFFFFFF80000048, crypto map: Tunnel1-head-0
sa timing: remaining key lifetime (k/sec): (32/1605)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:

Observe el flow_id en la salida, debe coincidir con el ID de flujo instalado en el plano de reenvío.

Plano de control PD

Estadísticas entre IOSd y el plano de control PD

<#root>

C9300X#

show platfor software ipsec policy statistics

PAL CMD	REQUEST	REPLY OK	REPLY ERR	ABORT
SADB_INIT_START	3	3	0	0
SADB_INIT_COMPLETED	3	3	0	0
SADB_DELETE	2	2	0	0
SADB_ATTR_UPDATE	4	4	0	0
SADB_INTF_ATTACH	3	3	0	0
SADB_INTF_UPDATE	0	0	0	0
SADB_INTF_DETACH	2	2	0	0
ACL_INSERT	4	4	0	0
ACL_MODIFY	0	0	0	0
ACL_DELETE	3	3	0	0
PEER_INSERT	7	7	0	0
PEER_DELETE	6	6	0	0
SPI_INSERT	39	37	2	0
SPI_DELETE	36	36	0	0
CFLOW_INSERT	5	5	0	0
CFLOW_MODIFY	33	33	0	0
CFLOW_DELETE	4	4	0	0
IPSEC_SA_DELETE	76	76	0	0
TBAR_CREATE	0	0	0	0
TBAR_UPDATE	0	0	0	0
TBAR_REMOVE	0	0	0	0
	0	0	0	0
PAL NOTIFY	RECEIVE	COMPLETE	PROC ERR	IGNORE

NOTIFY_RP	0	0	0	0
SA_DEAD	0	0	0	0
SA_SOFT_LIFE	46	46	0	0
IDLE_TIMER	0	0	0	0
DPD_TIMER	0	0	0	0
INVALID_SPI	0	0	0	0
	0	5	0	0
VTI SADB	0	33	0	0
TP SADB	0	40	0	0

IPSec PAL database summary:

DB NAME	ENT ADD	ENT DEL	ABORT
PAL_SADB	3	2	0
PAL_SADB_ID	3	2	0
PAL_INTF	3	2	0
PAL_SA_ID	76	74	0
PAL_ACL	0	0	0
PAL_PEER	7	6	0
PAL_SPI	39	38	0
PAL_CFLOW	5	4	0
PAL_TBAR	0	0	0

Tabla de objetos SADB

<#root>

C9300X#

show plat software ipsec switch active f0 sadb all

IPsec SADB object table:

SADB-ID	Hint	Complete	#RefCnt	#CfgCnt	#ACL-Ref
3	vir-tun-int	true	2	0	0

entrada SADB

<#root>

C9300X#

show plat software ipsec switch active f0 sadb identifier 3

```

===== SADB id: 3
      hint: vir-tun-int
    completed: true
reference count: 2
configure count: 0
  ACL reference: 0

```

```

SeqNo (Static/Dynamic)      ACL id
-----

```

Información de flujo de IPsec

<#root>

C9300X#

```
show plat software ipsec switch active f0 flow all
```

=====

Flow id: 97

```
        mode: tunnel
        direction: outbound
        protocol: esp
           SPI: 0x42709657
    local IP addr: 198.51.100.1
    remote IP addr: 192.0.2.2
    crypto map id: 0
           SPD id: 3
        cpp SPD id: 0
    ACE line number: 0
        QFP SA handle: INVALID
    crypto device id: 0
    IOS XE interface id: 65
        interface name: Tunnel1
        use path MTU: FALSE
        object state: active
    object bind state: new
```

=====

Flow id: 98

```
        mode: tunnel
        direction: inbound
        protocol: esp
           SPI: 0x4fe26715
    local IP addr: 198.51.100.1
    remote IP addr: 192.0.2.2
    crypto map id: 0
           SPD id: 3
        cpp SPD id: 0
    ACE line number: 0
        QFP SA handle: INVALID
    crypto device id: 0
    IOS XE interface id: 65
        interface name: Tunnel1
        object state: active
```

Troubleshoot

IOSd

Estos comandos debug y show se recolectan comúnmente:

```
<#root>
```

```
show crypto eli all
```

```
show crypto socket
```

```
show crypto map
```

```
show crypto ikev2 sa detail
```

```
show crypto ipsec sa
```

```
show crypto ipsec internal
```

```
<#root>
```

```
debug crypto ikev2
```

```
debug crypto ikev2 error
```

```
debug crypto ikev2 packet
```

```
debug crypto ipsec
```

```
debug crypto ipsec error
```

```
debug crypto kmi
```

```
debug crypto socket
```

```
debug tunnel protection
```

Plano de control PD

Para verificar las operaciones del plano de control PD, utilice los pasos de verificación mostrados anteriormente. Para depurar cualquier problema relacionado con el plano de control PD, habilite los debugs del plano de control PD:

1. Aumente el nivel de registro btrace a verbose:

```
<#root>
```

```
C9300X#
```

```
set platform software trace forwarding-manager switch active f0 ipsec verbose
```

```
C9300X#
```

```
show platform software trace level forwarding-manager switch active f0 | in ipsec
```

```
ipsec
```

```
Verbose
```

2. Habilite la depuración condicional del plano de control PD:

```
<#root>
```

```
C9300X#
```

```
debug platform condition feature ipsec controlplane submode level verbose
```

```
C9300X#
```

```
show platform conditions
```

```
Conditional Debug Global State: Stop
```

Feature	Type	Submode	Level
IPSEC		controlplane	N/A

```
verbose
```

3. Recopile la salida de debug de la salida de btrace fman_fp:

```
<#root>
```

```
C9300X#
```

```
show logging process fman_fp module ipsec internal
```

```
Logging display requested on 2022/10/19 20:57:52 (UTC) for Hostname: [C9300X], Model: [C9300X-24Y], Ver
```

```
Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds  
executing cmd on chassis 1 ...
```

```
Unified Decoder Library Init .. DONE
```

```
Found 1 UTF Streams
```

2022/10/19 20:50:36.686071658 {fman_fp_F0-0}{1}: [ipsec] [22441]: (ERR): IPSEC-PAL-IB-Key::
2022/10/19 20:50:36.686073648 {fman_fp_F0-0}{1}: [ipsec] [22441]: (ERR): IPSEC-b0 d0 31 04 85 36 a6 08

Plano de datos PD

Verificar las estadísticas del túnel IPsec del plano de datos, incluidas las caídas IPsec comunes como HMAC o los fallos de reproducción

<#root>

C9300X#

show platform software fed sw active ipsec counters if-id all

#####

Flow Stats for if-id 0x41

#####

Inbound Flow Info for

flow id: 98

SA Index: 1

Asic Instance 0: SA Stats

Packet Format Check Error:	0
Invalid SA:	0
Auth Fail:	0
Sequence Number Overflows:	0
Anti-Replay Fail:	0
Packet Count:	200
Byte Count:	27600

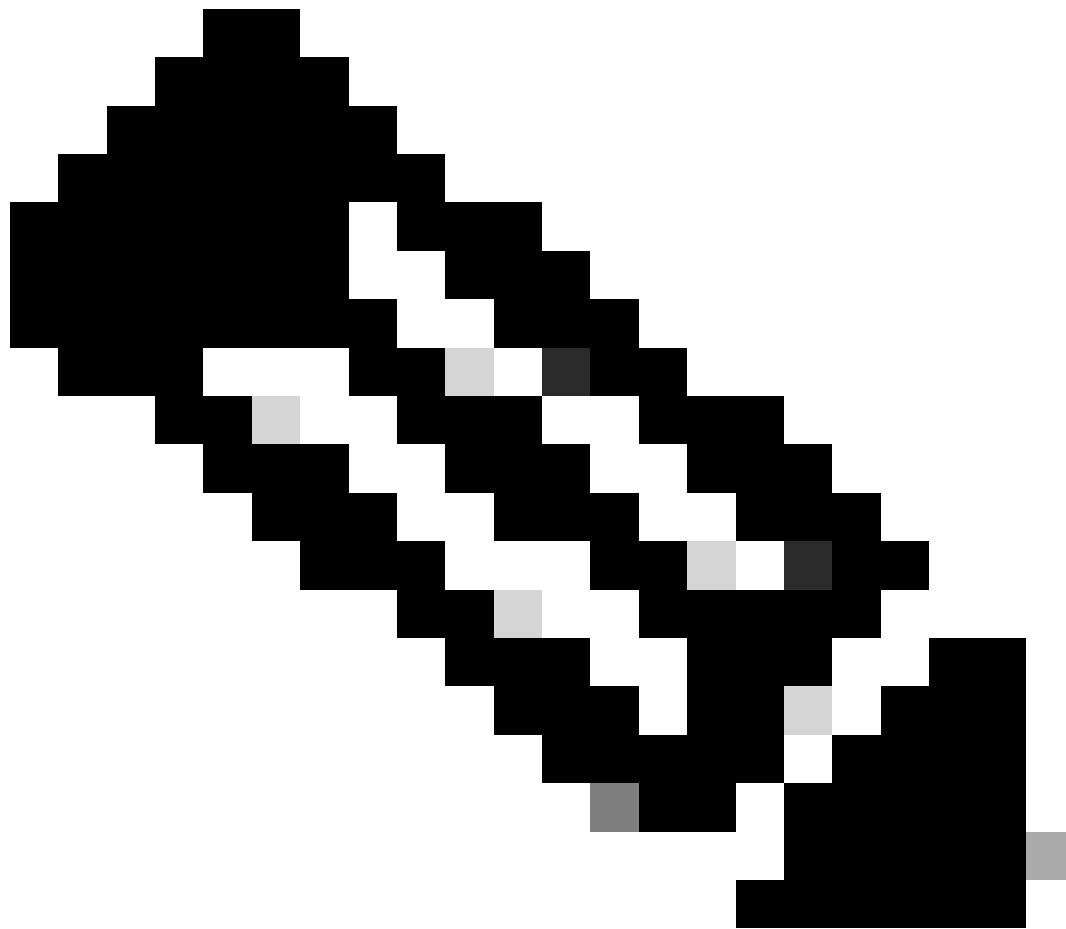
Outbound Flow Info for

flow id: 97

SA Index: 1025

Asic Instance 0: SA Stats

Packet Format Check Error:	0
Invalid SA:	0
Auth Fail:	0
Sequence Number Overflows:	0
Anti-Replay Fail:	0
Packet Count:	200
Byte Count:	33600



Nota: el id de flujo coincide con el id de flujo en la salida show crypto ipsec sa. Las estadísticas de flujo individuales también se pueden obtener con el comando show platform software fed switch active ipsec counters sa <sa_id>, donde sa_id cambia el índice SA en la salida anterior.

Dataplane Packet-tracer

Packet-tracer en la plataforma UADP ASIC se comporta de manera muy diferente que en el sistema basado en QFP. Se puede habilitar con un disparador manual o un disparador basado en PCAP. A continuación se muestra un ejemplo del uso del desencadenador basado en PCAP (EPC).

1. Habilite EPC e inicie la captura:

```
<#root>
```

```
C9300X#
```

```
monitor capture test interface twentyFiveGigE 1/0/2 in match ipv4 10.1.1.2/32 any
```

```
<#root>
```

```
C9300X#
```

```
show monitor capture test
```

```
Status Information for Capture test
```

```
Target Type:
```

```
Interface: TwentyFiveGigE1/0/2, Direction: IN
```

```
Status : Inactive
```

```
Filter Details:
```

```
IPv4
```

```
Source IP: 10.1.1.2/32
```

```
Destination IP: any
```

```
Protocol: any
```

```
Buffer Details:
```

```
Buffer Type: LINEAR (default)
```

```
Buffer Size (in MB): 10
```

```
File Details:
```

```
File not associated
```

```
Limit Details:
```

```
Number of Packets to capture: 0 (no limit)
```

```
Packet Capture duration: 0 (no limit)
```

```
Packet Size to capture: 0 (no limit)
```

```
Maximum number of packets to capture per second: 1000
```

```
Packet sampling rate: 0 (no sampling)
```

2. Ejecute el resto y detenga la captura:

```
<#root>
```

```
C9300X#
```

```
monitor capture test start
```

```
Started capture point : test
```

```
*Oct 18 18:34:09.656: %BUFCAP-6-ENABLE: Capture Point test enabled.
```

```
<run traffic test>
```

```
C9300X#
```

```
monitor capture test stop
```

```
Capture statistics collected at software:
```

```
Capture duration - 23 seconds
```

```
Packets received - 5
```

```
Packets dropped - 0
```

```
Packets oversized - 0
```

```
Bytes dropped in ASIC - 0
```

```
Capture buffer will exist till exported or cleared
```

```
Stopped capture point : test
```

3. Exportar la captura a flash

<#root>

C9300X#

```
show monitor capture test buff
```

```
*Oct 18 18:34:33.569: %BUFCAP-6-DISABLE
```

```
Starting the packet display ..... Press Ctrl + Shift + 6 to exit
```

```
 1  0.000000    10.1.1.2 -> 10.2.1.2    ICMP 114 Echo (ping) request  id=0x0003, seq=0/0, ttl=255
 2  0.000607    10.1.1.2 -> 10.2.1.2    ICMP 114 Echo (ping) request  id=0x0003, seq=1/256, ttl=2
 3  0.001191    10.1.1.2 -> 10.2.1.2    ICMP 114 Echo (ping) request  id=0x0003, seq=2/512, ttl=2
 4  0.001760    10.1.1.2 -> 10.2.1.2    ICMP 114 Echo (ping) request  id=0x0003, seq=3/768, ttl=2
 5  0.002336    10.1.1.2 -> 10.2.1.2    ICMP 114 Echo (ping) request  id=0x0003, seq=4/1024, ttl=
```

C9300X#

```
monitor capture test export location flash:test.pcap
```

4. Ejecute packet-tracer:

<#root>

C9300X#

```
show platform hardware fed switch 1 forward interface TwentyFiveGigE 1/0/2 pcap flash:test.pcap number 1
```

```
Show forward is running in the background. After completion, syslog will be generated.
```

C9300X#

```
*Oct 18 18:36:56.288: %SHFWD-6-PACKET_TRACE_DONE: Switch 1 F0/0: fed: Packet Trace Complete: Execute (
```

```
*Oct 18 18:36:56.288: %SHFWD-6-PACKET_TRACE_FLOW_ID: Switch 1 F0/0: fed: Packet Trace Flow id is 131077
```

C9300X#

```
C9300X#show plat hardware fed switch 1 forward last summary
```

```
Input Packet Details:
```

```
###[ Ethernet ]###
```

```
dst      = b0:8b:d0:8d:6b:d6
```

```
src=78:ba:f9:ab:a7:03
```

```
type     = 0x800
```

```
###[ IP ]###
```

```
version  = 4
```

```
ihl      = 5
```

```
tos      = 0x0
```

```
len      = 100
```

```
id       = 15
```

```
flags    =
```

```
frag     = 0
```

```
ttl      = 255
```

```
proto    = icmp
```

```
chksum   = 0xa583
```

```
src=10.1.1.2
```

```
dst      = 10.2.1.2
```

```
options  = ''
```

```
###[ ICMP ]###
```

```
type     = echo-request
```

```
code     = 0
```


chksum = 0xae17
id = 0x3
seq = 0x0

###[Raw]###

load = '00 00 00 00 01 1B CF 14 AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD AB CD A

Ingress:

Port : TwentyFiveGigE1/0/2
Global Port Number : 2
Local Port Number : 2
Asic Port Number : 1
Asic Instance : 1
Vlan : 4095
Mapped Vlan ID : 1
STP Instance : 1
BlockForward : 0
BlockLearn : 0
L3 Interface : 38
IPv4 Routing : enabled
IPv6 Routing : enabled
Vrf Id : 0

Adjacency:

Station Index : 179
Destination Index : 20754
Rewrite Index : 24
Replication Bit Map : 0x1 ['remoteData']

Decision:

Destination Index : 20754 [DI_RCP_PORT3]
Rewrite Index : 24
Dest Mod Index : 0 [IGR_FIXED_DMI_NULL_VALUE]
CPU Map Index : 0 [CMI_NULL]
Forwarding Mode : 3 [Other or Tunnel]
Replication Bit Map : ['remoteData']
Winner : L3FWDIPV4_LOOKUP
Qos Label : 1
SGT : 0
DGTID : 0

Egress:

Possible Replication :
Port : RCP
Asic Instance : 0
Asic Port Number : 0
Output Port Data :
Port : RCP
Asic Instance : 0
Asic Port Number : 90
Unique RI : 0
Rewrite Type : 0 [Unknown]
Mapped Rewrite Type : 229 [IPSEC_TUNNEL_MODE_ENCAP_FIRSTPASS_OUTERV4_INNERV4]
Vlan : 0
Mapped Vlan ID : 0
RCP, mappedRii.fdMuxProfileSet = 1 , get fdMuxProfile from MappedRii
Qos Label : 1
SGT : 0

Input Packet Details:

N/A: Recirculated Packet

Ingress:

Port : Recirculation Port
Asic Port Number : 90
Asic Instance : 0
Vlan : 0
Mapped Vlan ID : 2

```

STP Instance          : 0
BlockForward         : 0
BlockLearn           : 0
L3 Interface         : 38
    IPv4 Routing      : enabled
    IPv6 Routing      : enabled
    Vrf Id            : 0
Adjacency:
    Station Index     : 177
    Destination Index : 21304
    Rewrite Index     : 21
    Replication Bit Map : 0x1    ['remoteData']
Decision:
    Destination Index : 21304
    Rewrite Index     : 21
    Dest Mod Index    : 0        [IGR_FIXED_DMI_NULL_VALUE]
    CPU Map Index     : 0        [CMI_NULL]
    Forwarding Mode   : 3        [Other or Tunnel]
    Replication Bit Map :        ['remoteData']
    Winner            :          L3FWDIPV4_LOOKUP
    Qos Label         : 1
    SGT               : 0
    DGTID             : 0

```

```

Egress:
    Possible Replication :
        Port             : TwentyFiveGigE1/0/1
    Output Port Data    :
        Port             : TwentyFiveGigE1/0/1
        Global Port Number : 1
        Local Port Number  : 1
        Asic Port Number   : 0
        Asic Instance     : 1
        Unique RI          : 0
        Rewrite Type       : 0        [Unknown]
        Mapped Rewrite Type : 13    [L3_UNICAST_IPV4_PARTIAL]
        Vlan               : 0
        Mapped Vlan ID    : 0

```

```

Output Packet Details:
    Port             : TwentyFiveGigE1/0/1

```

```

###[ Ethernet ]###
dst      = 00:62:ec:da:e0:02
src=b0:8b:d0:8d:6b:e4
type     = 0x800

```

```

###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 168
id       = 2114
flags    = DF
frag     = 0
ttl      = 254
proto    = ipv6_crypt
chksum   = 0x45db
src=198.51.100.1
dst      = 192.0.2.2
options  = ''

```

```

###[ Raw ]###      load      = '

```

```

6D 18 45 C9

```

```

00 00 00 06 09 B0 DC 13 11 FA DC F8 63 98 51 98 33 11 9C C0 D7 24 BF C2 1C 45 D3 1B 91 0B 5F B4 3A C0

```

C9300X#

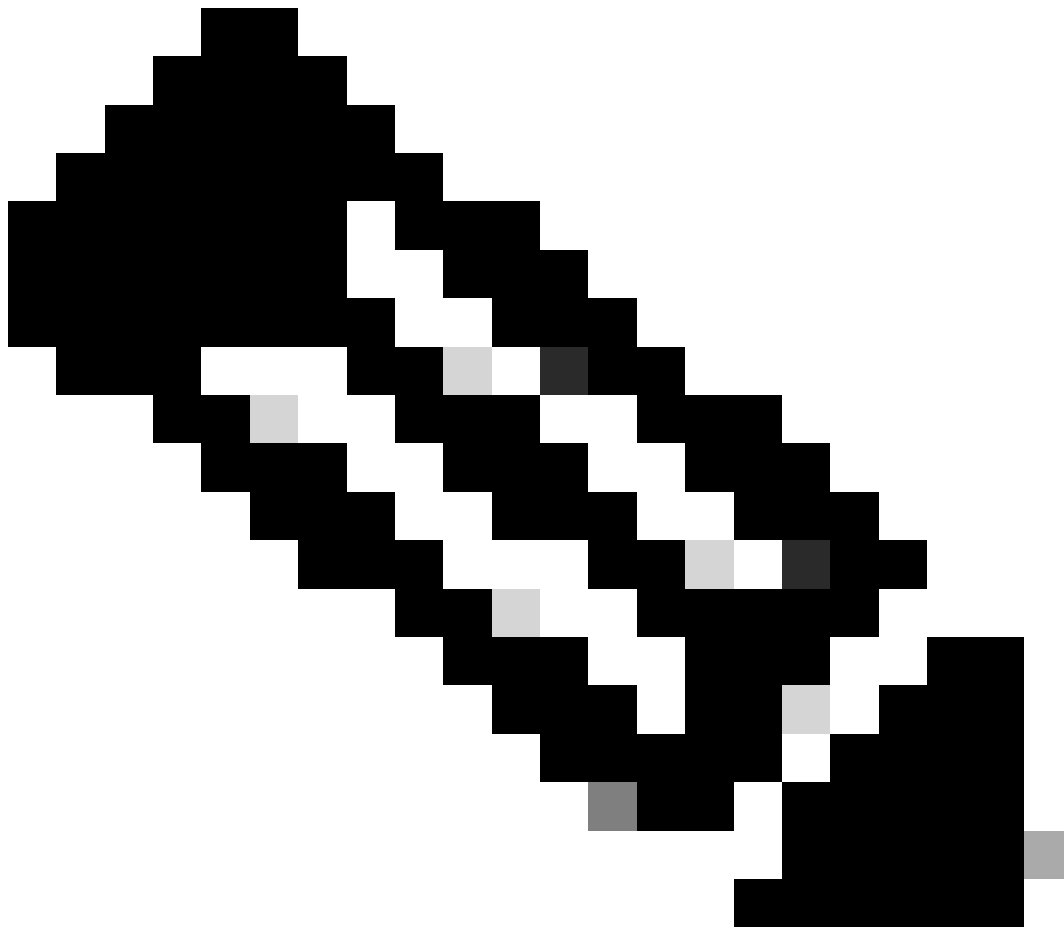
show crypto ipsec sa | in current outbound

current outbound spi:

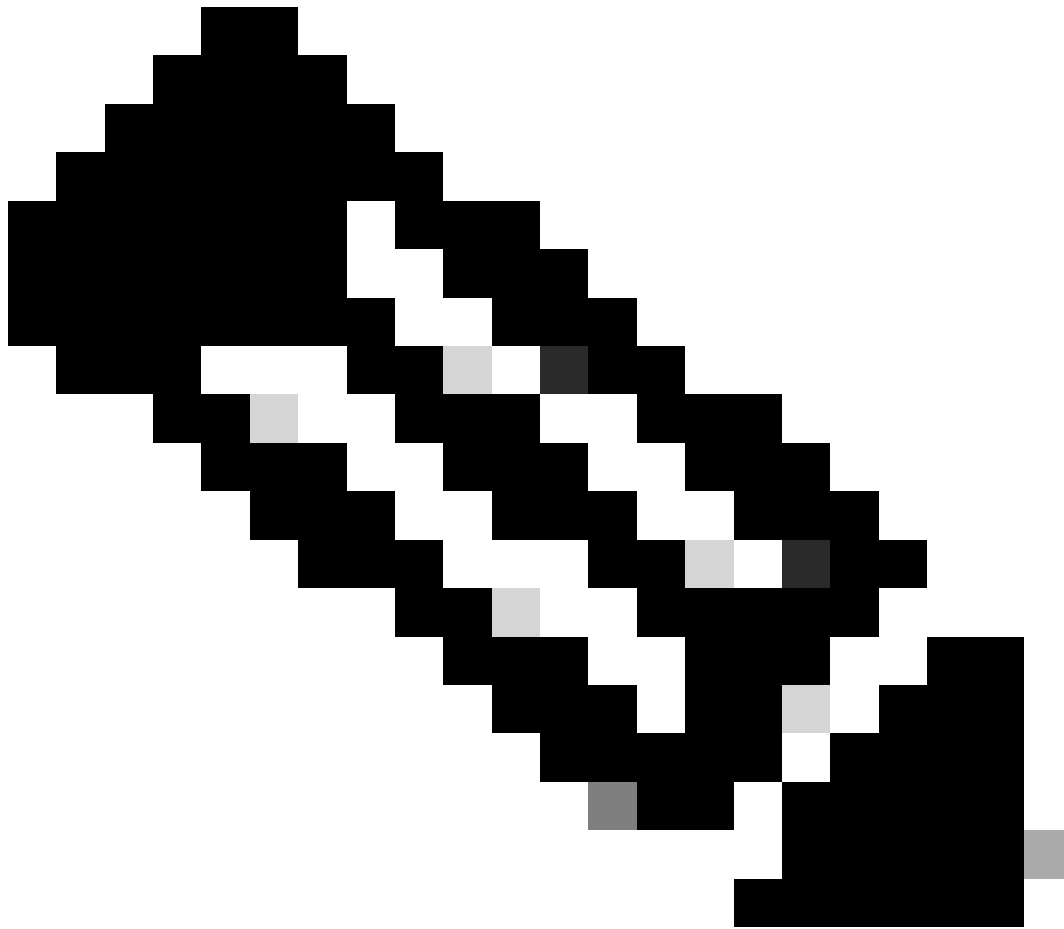
0x6D1845C9

(1830307273)

<-- Matches the load result in packet trace



Nota: en la salida anterior, la salida reenviada del paquete es el paquete ESP con el SPI de SA saliente actual. Para un análisis más detallado de la decisión de reenvío de la FED, la variante detail del mismo comando. Ejemplo: se puede utilizar show plat hardware fed switch 1 forward last detail.



Nota: La depuración del plano de datos de PD solo se debe habilitar con la ayuda del TAC. Se trata de seguimientos de muy bajo nivel que la ingeniería necesita si el problema no se puede identificar a través de CLI/depuraciones normales.

<#root>

C9300X#

```
set platform software trace fed switch active ipsec verbose
```

```
C9300X#
```

```
debug platform condition feature ipsec dataplane submode all level verbose
```

```
C9300X#
```

```
show logging process fed module ipsec internal
```

Depuraciones de IPsec PD SHIM

```
<#root>
```

```
debug platform software ipsec info
```

```
debug platform software ipsec error
```

```
debug platform software ipsec verbose
```

```
debug platform software ipsec all
```

Información Relacionada

- [Configuración de IPsec en switches Catalyst 9300](#)

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).