Resolución de Problemas del Túnel de Radio a Radio de la Fase 2 DMVPN

Contenido

Introducción

Este documento describe cómo resolver problemas de un túnel DMVPN de radio a radio de fase 2 cuando no establece.

Prerequisites

Requirements

Cisco recomienda que tenga conocimientos sobre los siguientes temas:

- Red privada virtual multipunto dinámica (DMVPN)
- Protocolos IKE/IPSEC
- Protocolo de resolución de salto siguiente (NHRP)

Componentes Utilizados

Este documento se basa en esta versión de software:

- Cisco CSR1000V (VXE), versión 17.03.08

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

Este documento describe cómo configurar y utilizar diferentes herramientas de troubleshooting en un problema DMVPN común. El problema es la negociación fallida de un túnel DMVPN de fase 2, en el que el estado de la DMVPN es de origen y aparece UP con la asignación correcta de multiacceso sin difusión (NBMA)/túnel al spoke de destino. Sin embargo, en el spoke de destino se muestra una asignación incorrecta.

Antecedentes teóricos

Es importante comprender cómo se establecen los túneles de radio a radio al configurar DMVPN Phase 2. Esta sección proporciona un breve resumen teórico del proceso NHRP durante esta fase.

En la fase 2 de DMVPN puede crear túneles dinámicos de radio a radio a demanda. Esto es posible porque, en todos los dispositivos de la nube DMVPN (hub y radios), el modo de la interfaz de túnel cambia a multipunto de encapsulación de routing genérico (GRE). Una de las características clave de esta fase es que los demás dispositivos no perciben el hub como el salto siguiente. En cambio, todos los radios tienen la información de ruteo de cada uno. Al establecer un túnel de radio a radio en la fase 2, se activa un proceso NHRP donde los radios aprenden la información sobre otros radios y realizan un mapeo entre la NBMA y las direcciones IP del túnel.

Los siguientes pasos enumeran cómo se activa el proceso de resolución NHRP:

1. Cuando el spoke de origen intenta alcanzar la LAN del spoke de destino, realiza una búsqueda de ruta que activa el mensaje de solicitud de resolución para obtener la dirección NBMA del spoke de destino. El spoke de origen envía este mensaje inicial al hub.

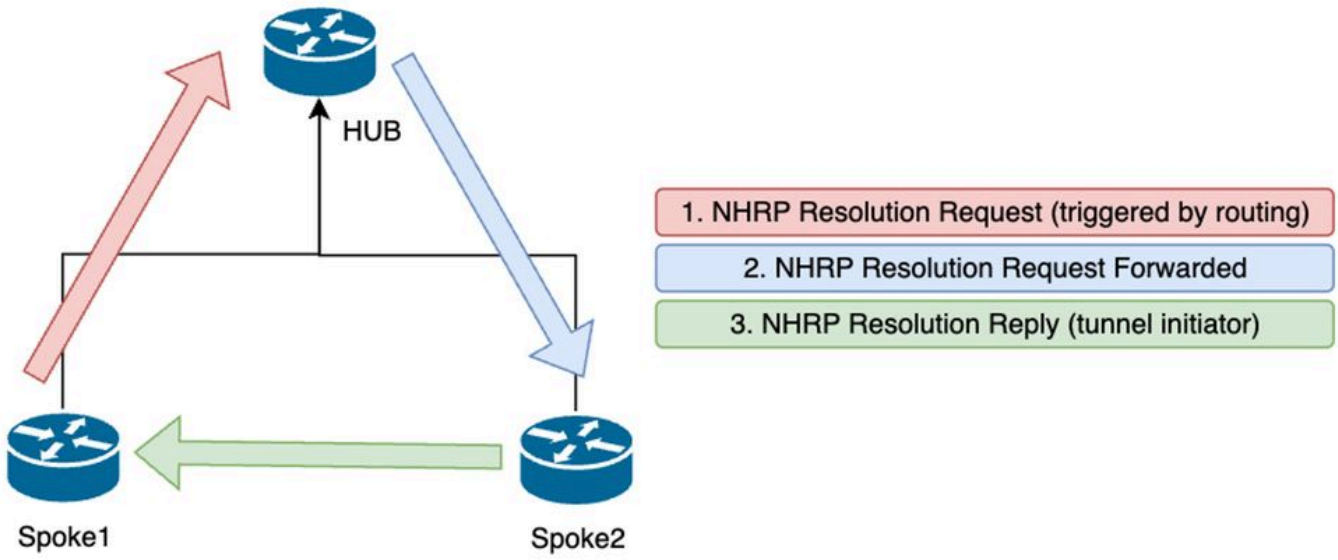
2. El hub recibe la solicitud de resolución y la reenvía al spoke de destino.

3. El spoke de destino envía la respuesta de resolución al spoke de origen. Si la configuración del túnel tiene un perfil IPSEC vinculado:

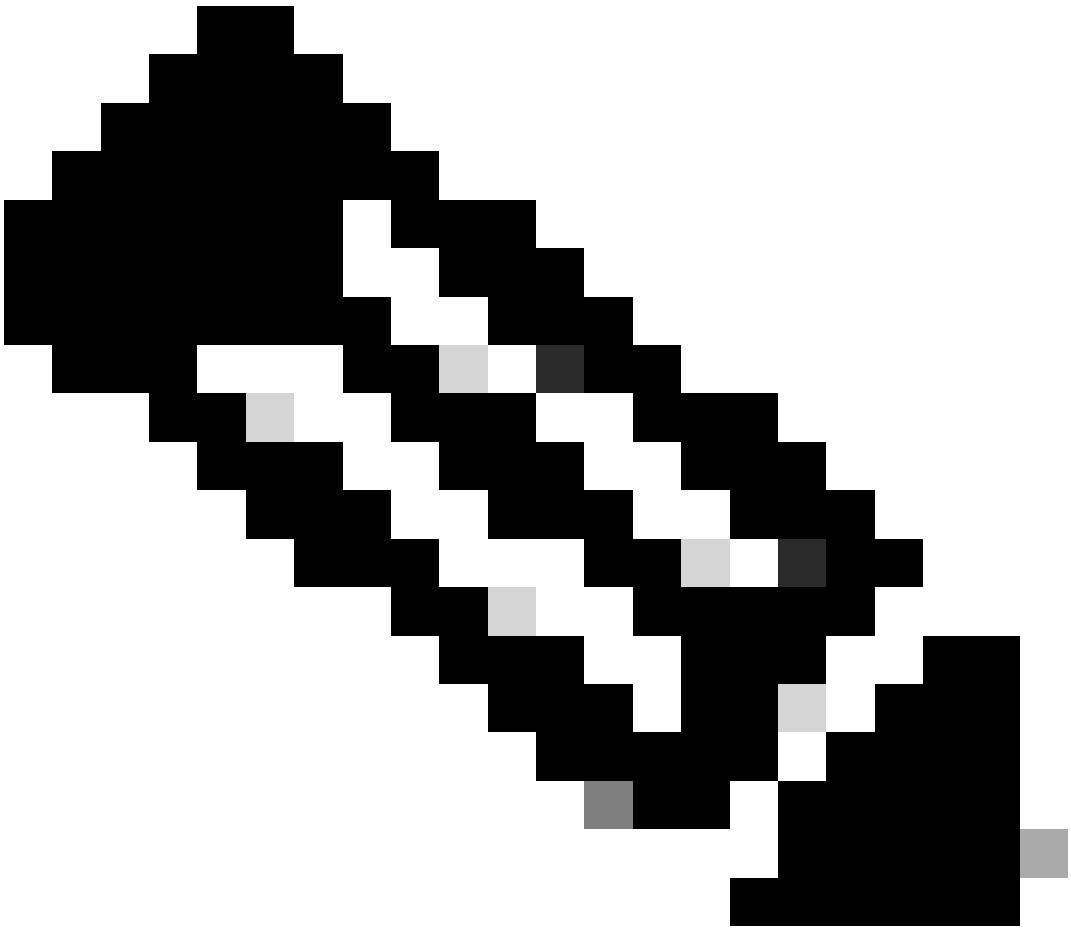
    - El proceso de resolución NHRP se retrasa hasta que los protocolos IKE/IPSEC puedan establecerse.

    - El spoke de destino inicia y establece los túneles IKE/IPSEC.

    - A continuación, se reanuda el proceso NHRP y el spoke de destino envía la respuesta de resolución al spoke de origen utilizando el túnel IPSEC como método de transporte.



Flujo de mensajes NHRP entre radios en la fase 2



Nota: Antes de que pueda iniciarse el proceso de resolución, todos los radios deben estar

---

ya registrados en el HUB.

Topología

Este diagrama muestra la topología utilizada para el escenario:

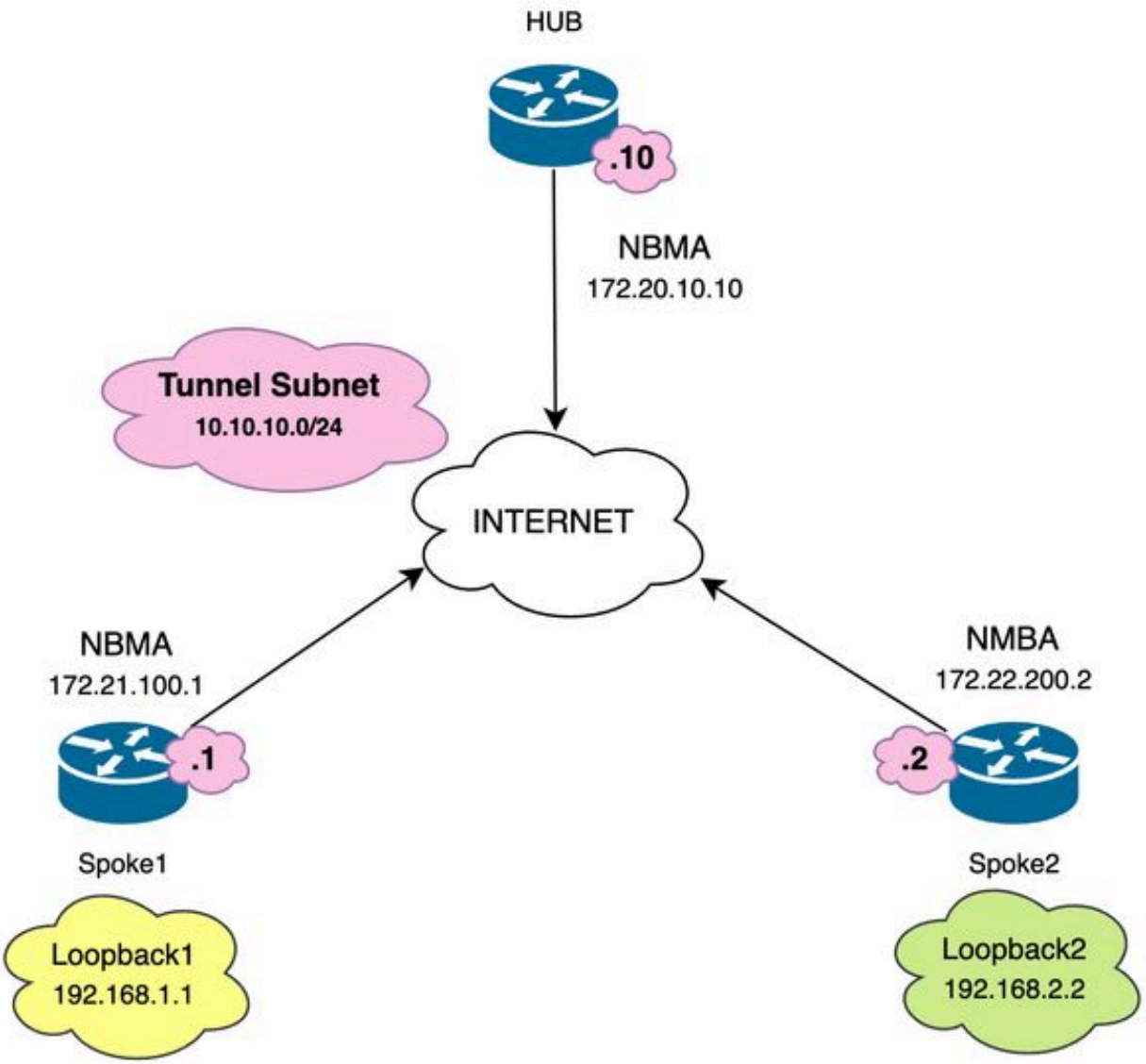


Diagrama de red y subredes IP utilizadas

Pasos para la resolución de problemas

En esta situación, no se establece el túnel de radio a radio entre Spoke1 y Spoke2, lo que afecta a la comunicación entre sus recursos locales (representados por interfaces de loopback), ya que no pueden comunicarse entre sí.

```
SPOKE1#ping 192.168.2.2 source loopback1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Validación inicial

Al encontrarse con un escenario de este tipo, es importante comenzar por validar la configuración del túnel y asegurarse de que ambos dispositivos tienen los valores correctos dentro de él. Para revisar la configuración del túnel, ejecute el comando show running-config interface tunnel<ID>.

Configuración del túnel Spoke 1:

<#root>

```
SPOKE1#show running-config interface tunnel10
Building configuration...

Current configuration : 341 bytes
!
interface Tunnel10
ip address 10.10.10.1 255.255.255.0
no ip redirects
ip nhrp authentication DMVPN


ip nhrp map 10.10.10.10 172.20.10.10


ip nhrp map multicast 172.20.10.10


ip nhrp network-id 10
ip nhrp nhs 10.10.10.10


tunnel source GigabitEthernet1
tunnel mode gre multipoint


tunnel protection IPSEC profile IPSEC_Profile_1


end
```

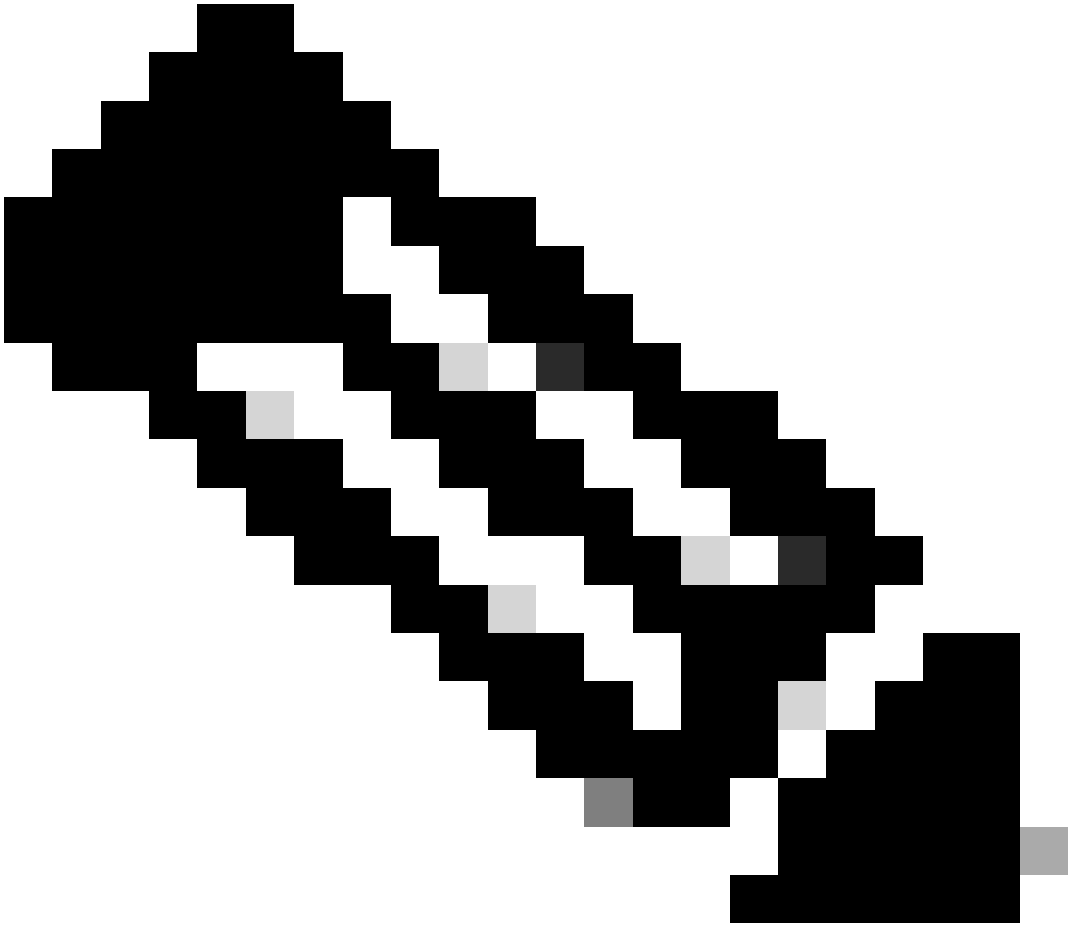
Configuración del túnel Spoke 2:

<#root>

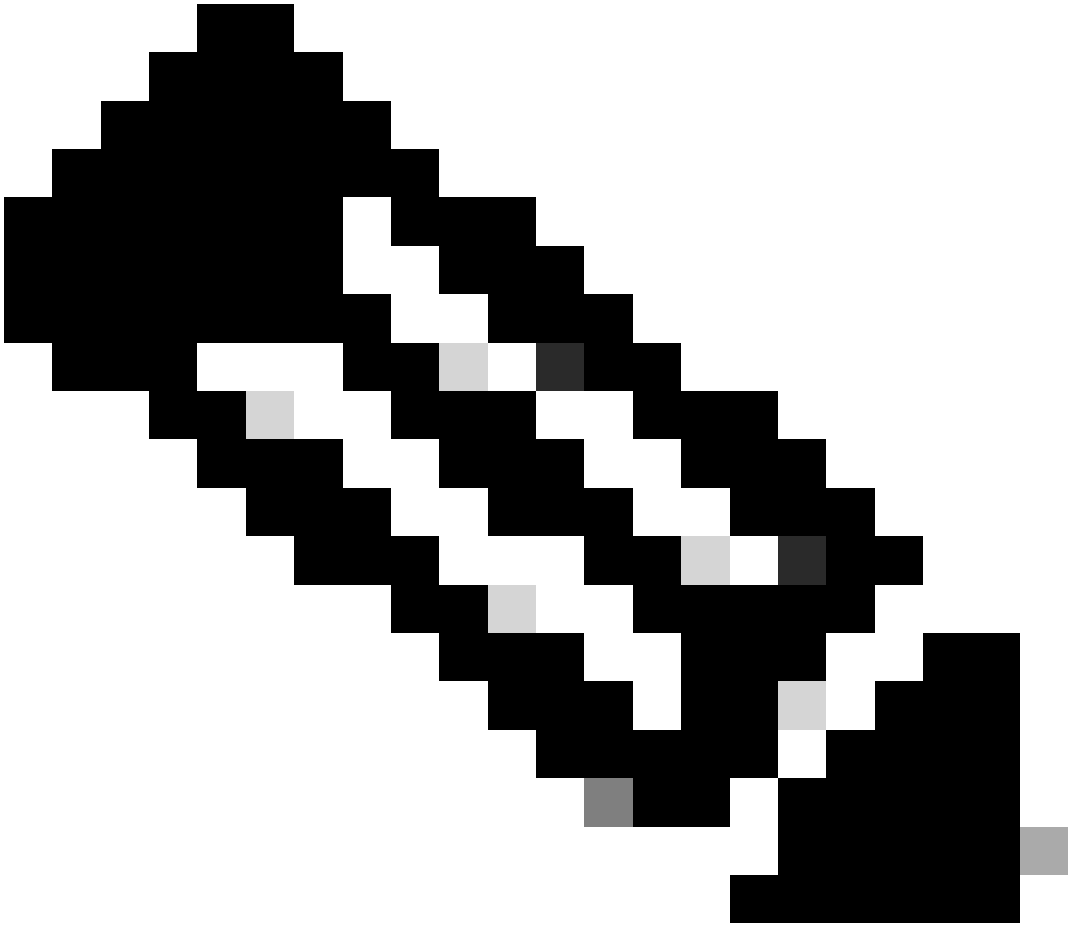
```
SPOKE2#show running-config interface tunnel10
Building configuration...

Current configuration : 341 bytes
!
interface Tunnel10
ip address 10.10.10.2 255.255.255.0
no ip redirects
ip nhrp authentication DMVPN


ip nhrp map 10.10.10.10 172.20.10.10


ip nhrp map multicast 172.20.10.10


ip nhrp network-id 10
ip nhrp nhs 10.10.10.10


tunnel source GigabitEthernet1
tunnel mode gre multipoint


tunnel protection IPSEC profile IPSEC_Profile_1


end
```

En la configuración que necesita para validar que la asignación al HUB es correcta, la cadena de autenticación NHRP coincide entre los dispositivos, ambos radios tienen configurada la misma fase DMVPN y, si se utiliza protección IPSEC, verifique que se aplica la configuración criptográfica correcta.

Si la configuración es correcta e incluye protección IPSEC, es necesario comprobar que los protocolos IKE e IPSEC funcionan correctamente. Esto se debe a que NHRP utiliza el túnel IPSEC como método de transporte para negociar completamente. Para verificar el estado de los protocolos IKE/IPSEC ejecute el comando show crypto IPSEC sa peer x.x.x.x (donde x.x.x.x es la dirección IP NBMA del spoke con el que intenta establecer el túnel).



Nota: para comprobar si el túnel IPSEC está activo, la sección de carga de seguridad de encapsulación (ESP) entrante y saliente debe tener la información del túnel (SPI, conjunto de transformación, etc.). Todos los valores mostrados en esta sección deben coincidir en ambos extremos.



Nota: si se identifica algún problema con IKE/IPSEC, la resolución de problemas debe centrarse en esos protocolos.

Estado del túnel IKE/IPSEC en Spoke1:

<#root>

SPOKE1#

show crypto IPSEC sa peer 172.22.200.2

```
interface: Tunnel10
Crypto map tag: Tunnel10-head-0, local addr 172.21.100.1

protected vrf: (none)
local ident (addr/mask/prot/port): (172.21.100.1/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.22.200.2/255.255.255.255/47/0)
current_peer 172.22.200.2 port 500
PERMIT, flags={origin_is_acl,}
```

#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0

#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0

#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.: 172.21.100.1, remote crypto endpt.: 172.22.200.2
plaintext mtu 1458, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
current outbound spi: 0x6F6BF94A(1869347146)
PFS (Y/N): N, DH group: none

inbound esp sas:

spi: 0x84502A19(2219846169)

transform: esp-256-aes esp-sha256-hmac

 ,
in use settings ={Transport, }
conn id: 2049, flow_id: CSR:49, sibling_flags FFFFFFFF80000008, crypto map: Tunnel10-head-0
sa timing: remaining key lifetime (k/sec): (4608000/28716)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:

spi: 0x6F6BF94A(1869347146)

transform: esp-256-aes esp-sha256-hmac

 ,
in use settings ={Transport, }
conn id: 2050, flow_id: CSR:50, sibling_flags FFFFFFFF80000008, crypto map: Tunnel10-head-0
sa timing: remaining key lifetime (k/sec): (4608000/28716)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:

Estado del túnel IKE/IPSEC en Spoke2:

<#root>

SPOKE2#

show crypto IPSEC sa peer 172.21.100.1

interface: Tunnel10
Crypto map tag: Tunnel10-head-0, local addr 172.22.200.2

protected vrf: (none)
local ident (addr/mask/prot/port): (172.22.200.2/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.21.100.1/255.255.255.255/47/0)
current_peer 172.21.100.1 port 500
PERMIT, flags={origin_is_acl,}

#pkts encaps: 16, #pkts encrypt: 16, #pkts digest: 16

#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0

#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.: 172.22.200.2, remote crypto endpt.: 172.21.100.1
plaintext mtu 1458, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
current outbound spi: 0x84502A19(2219846169)
PFS (Y/N): N, DH group: none

inbound esp sas:

spi: 0x6F6BF94A(1869347146)

transform: esp-256-aes esp-sha256-hmac ,

in use settings ={Transport, }
conn id: 2045, flow_id: CSR:45, sibling_flags FFFFFFFF80004008, crypto map: Tunnel10-head-0
sa timing: remaining key lifetime (k/sec): (4608000/28523)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:

```
spi: 0x84502A19(2219846169)


transform: esp-256-aes esp-sha256-hmac

 ,
in use settings ={Transport, }
conn id: 2046, flow_id: CSR:46, sibling_flags FFFFFFFF80004008, crypto map: Tunnel10-head-0
sa timing: remaining key lifetime (k/sec): (4607998/28523)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```

Las salidas muestran que en ambos radios el túnel IPSEC está activo, pero Spoke2 muestra paquetes cifrados (encaps) pero no paquetes descifrados (decaps). Mientras tanto, Spoke1 no muestra ningún paquete que fluya a través del túnel IPSEC. Esto indica que el problema puede estar en el protocolo NHRP.

Herramientas de solución de problemas

Después de realizar la validación inicial y corroborar la configuración y los protocolos IKE/IPSEC (si son necesarios) no están causando el problema de comunicación, puede utilizar las herramientas presentadas en esta sección para continuar con la solución de problemas.

Comandos útiles

El comando show dmvpn interface tunnel<ID> proporciona información de sesión específica de DMVPN (direcciones IP de túnel/NBMA, estado del túnel, tiempo de actividad/inactividad y atributo). Puede utilizar la palabra clave detail para mostrar los detalles de la sesión/socket criptográfico. Es importante mencionar que el estado del túnel debe coincidir en ambos extremos.

Spoke 1 show dmvpn interface tunnel<ID> output:

<#root>

```
SPOKE1#

show dmvpn interface tunnel10


Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
N - NATed, L - Local, X - No Socket
T1 - Route Installed, T2 - Nexthop-override, B - BGP
C - CTS Capable, I2 - Temporary
# Ent --> Number of NHRP entries with same NBMA peer
NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
UpDn Time --> Up or Down Time for a Tunnel
==========================================================================
```

```
Interface: Tunnel10, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,

# Ent Peer NBMA Addr Peer Tunnel Add  State UpDn Tm  Attrb
----- --------------- --------------- ----- -------- -----
   2
```

172.20.10.10 10.10.10.2 UP 00:00:51 I2

10.10.10.10 UP 02:53:27 S

Spoke 2 show dmvpn interface tunnel<ID> output:

<#root>

SPOKE2#

show dmvpn interface tunnel10

```
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
N - NATed, L - Local, X - No Socket
T1 - Route Installed, T2 - Nexthop-override, B - BGP
C - CTS Capable, I2 - Temporary
# Ent --> Number of NHRP entries with same NBMA peer
NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel10, IPv4 NHRP Details
Type:Spoke, NHRP Peers:2,

# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
----- --------------- --------------- ----- -------- -----
```

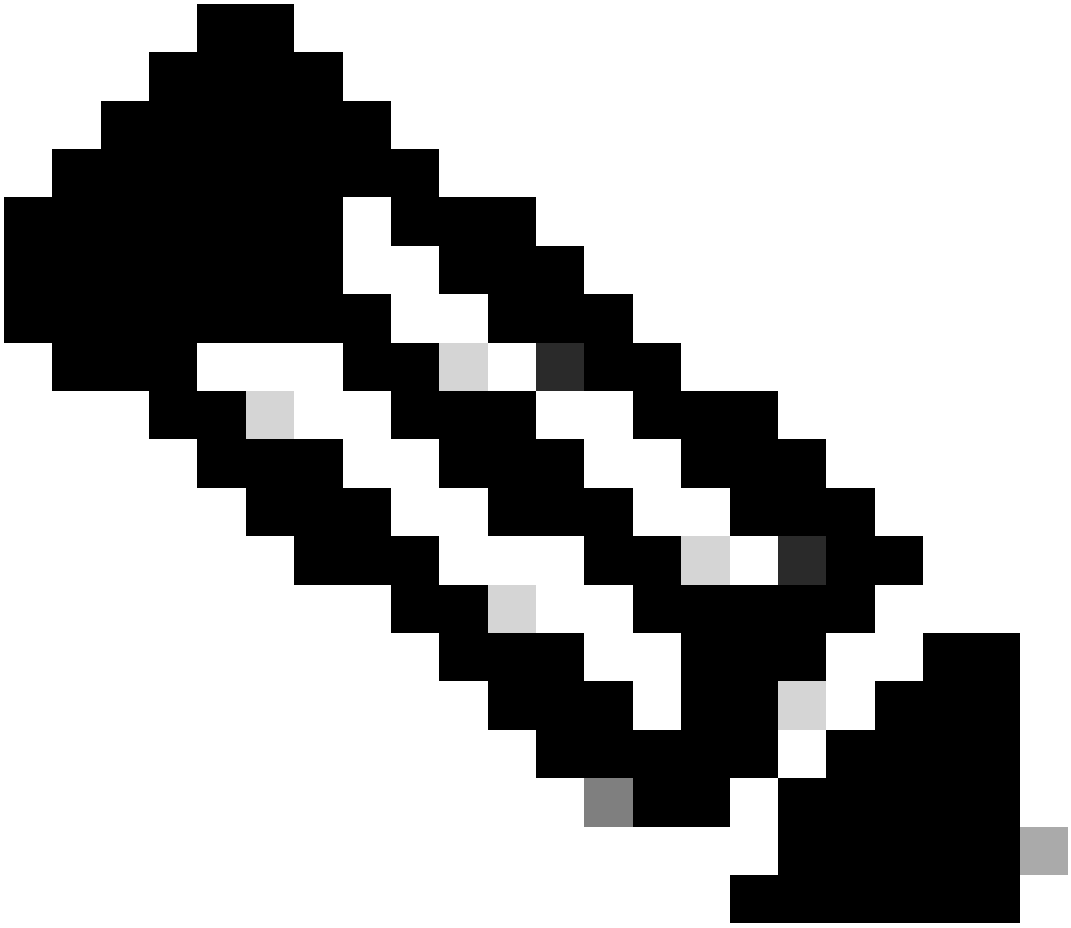
1 172.21.100.1 10.10.10.1 UP 00:03:53 D

1 172.20.10.10 10.10.10.10 UP 02:59:14 S

La salida de cada dispositivo muestra información diferente para cada radio. En la tabla Spoke1, puede ver que la entrada para Spoke 2 no incluye la dirección IP NBMA correcta y el atributo parece incompleto (I2). Por otro lado, la tabla Spoke2 muestra la asignación correcta (direcciones IP de NBMA/túnel) y el estado up que indica que el túnel se ha negociado completamente.

Los siguientes comandos pueden ser útiles durante el proceso de solución de problemas:

- show ip nhrp: Mostrar información de asignación NHRP
- show ip nhrp traffic interface tunnel10: Muestra estadísticas de tráfico NHRP

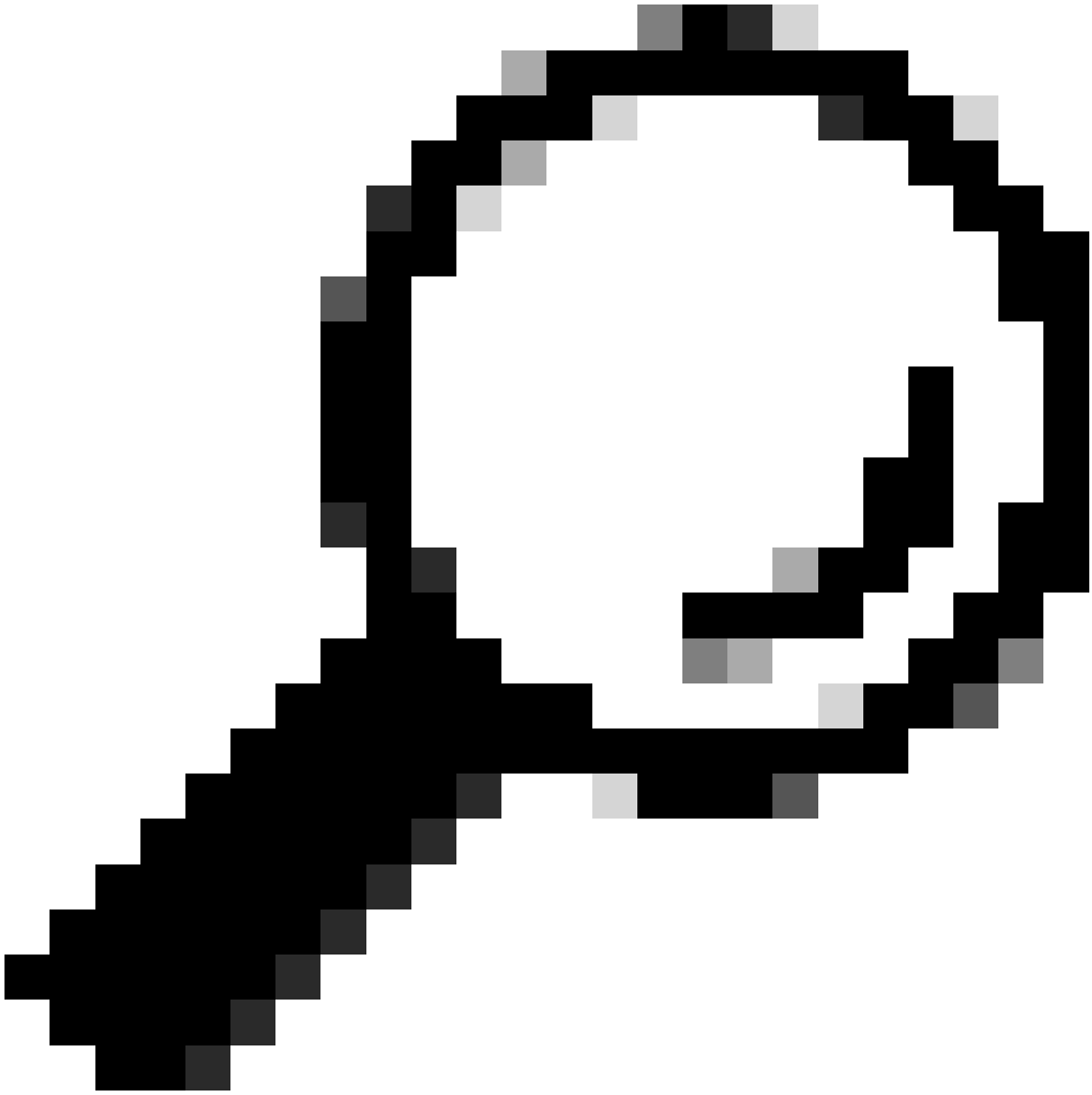


Nota: Para obtener información sobre las especificaciones de los comandos (sintaxis, descripción, palabras clave, ejemplo), consulte la Referencia de Comandos: Referencia de Comandos de Seguridad de Cisco IOS: Comandos S a Z

Depuraciones

Después de verificar la información previa y confirmar que el túnel está experimentando problemas de negociación, es necesario habilitar los debugs para observar cómo se intercambian los paquetes NHRP. Las siguientes depuraciones deben estar habilitadas en todos los dispositivos involucrados:

1. debug dmvpn condition peer NBMA x.x.x.x (donde x.x.x.x es la dirección IP del dispositivo remoto).
2. debug dmvpn all all: este comando habilita los comandos de depuración ISAKMP, IKEv2, IPSEC, DMVPN y NHRP.



Sugerencia: se recomienda utilizar el comando peer condition cada vez que habilite los debugs para que pueda ver la negociación de ese túnel específico.

Para ver el flujo NHRP completo, se usaron los siguientes comandos de depuración en cada dispositivo:

Spoke1

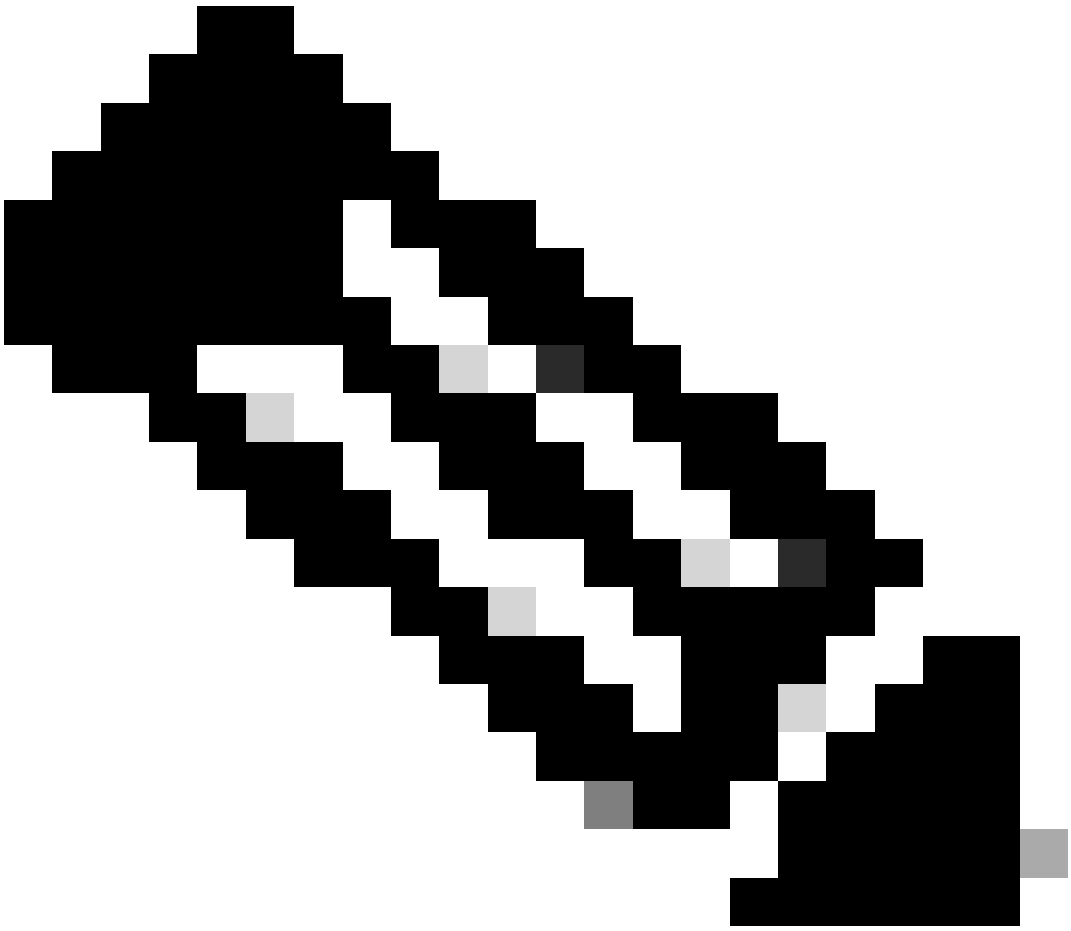
```
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```

HUB

```
debug dmvpn condition peer NBMA 172.21.100.1
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn all all
```

Spoke2

```
debug dmvpn condition peer NBMA 172.21.100.1
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```



Nota: Las depuraciones deben activarse y recopilarse simultáneamente en todos los

---

dispositivos involucrados.

Las depuraciones habilitadas en todos los dispositivos se muestran con el comando show debug:

<#root>

```
ROUTER#
show debug

IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:


Packet Infra debugs:

Ip Address Port
------------------------------------------------------|----------

NHRP:
NHRP protocol debugging is on
NHRP activity debugging is on
NHRP detail debugging is on
NHRP extension processing debugging is on
NHRP cache operations debugging is on
NHRP routing debugging is on
NHRP rate limiting debugging is on
NHRP errors debugging is on
NHRP events debugging is on

Cryptographic Subsystem:
Crypto ISAKMP debugging is on
Crypto ISAKMP Error debugging is on
Crypto IPSEC debugging is on
Crypto IPSEC Error debugging is on
Crypto secure socket events debugging is on
IKEV2:
IKEv2 error debugging is on
IKEv2 default debugging is on
IKEv2 packet debugging is on
IKEv2 packet hexdump debugging is on
IKEv2 internal debugging is on
Tunnel Protection Debugs:
Generic Tunnel Protection debugging is on

DMVPN:
DMVPN error debugging is on
DMVPN UP/DOWN event debugging is on
DMVPN detail debugging is on
DMVPN packet debugging is on
DMVPN all level debugging is on
```

Después de recolectar todos los debugs, debe comenzar a analizar los debugs en el spoke de origen (Spoke1), esto le permite rastrear la negociación desde el principio.

Resultado de depuración de Spoke1:

<#root>

```
---------------------- [IKE/IPSEC DEBUG OUTPUTS OMITTED]----------------------

*Feb 1 01:31:34.657: ISAKMP: (1016):

Old State = IKE_QM_R_QM2 New State = IKE_QM_PHASE2_COMPLETE


*Feb 1 01:31:34.657: IPSEC(key_engine): got a queue event with 1 KMI message(s)
*Feb 1 01:31:34.657: IPSEC(key_engine_enable_outbound): rec'd enable notify from ISAKMP
*Feb 1 01:31:34.657: CRYPTO_SS(TUNNEL SEC): Sending MTU Changed message
*Feb 1 01:31:34.661: IPSEC-IFC MGRE/Tu10(172.21.100.1/172.22.200.2): Got MTU message mtu 1458
*Feb 1 01:31:34.661: IPSEC-IFC MGRE/Tu10(172.21.100.1/172.22.200.2): connection lookup returned 80007F2
*Feb 1 01:31:34.662: CRYPTO_SS(TUNNEL SEC): Sending Socket Up message
*Feb 1 01:31:34.662: IPSEC-IFC MGRE/Tu10(172.21.100.1/172.22.200.2): connection lookup returned 80007F2
*Feb 1 01:31:34.662: IPSEC-IFC MGRE/Tu10(172.21.100.1/172.22.200.2):

tunnel_protection_socket_up


*Feb 1 01:31:34.662: IPSEC-IFC MGRE/Tu10(172.21.100.1/172.22.200.2): Signalling NHRP
*Feb 1 01:31:36.428: NHRP: Checking for delayed event NULL/10.10.10.2 on list (Tunnel10 vrf: global(0x0)
*Feb 1 01:31:36.429: NHRP: No delayed event node found.
*Feb 1 01:31:36.429: NHRP: There is no VPE Extension to construct for the request
*Feb 1 01:31:36.429: NHRP: Sending NHRP Resolution Request for dest: 10.10.10.2 to nexthop: 10.10.10.2
*Feb 1 01:31:36.429: NHRP: Attempting to send packet through interface Tunnel10 via DEST dst 10.10.10.2
*Feb 1 01:31:36.429: NHRP-DETAIL: First hop route lookup for 10.10.10.2 yielded 10.10.10.2, Tunnel10
*Feb 1 01:31:36.429: NHRP:

Send Resolution Request via Tunnel10 vrf: global(0x0), packet size: 85


*Feb 1 01:31:36.429: src: 10.10.10.1, dst: 10.10.10.2
*Feb 1 01:31:36.429: (F) afn: AF_IP(1), type: IP(800), hop: 255, ver: 1
*Feb 1 01:31:36.429: shtl: 4(NSAP), sstl: 0(NSAP)
*Feb 1 01:31:36.429: pktsz: 85 extoff: 52
*Feb 1 01:31:36.429: (M) flags: "router auth src-stable nat ",

reqid: 10


*Feb 1 01:31:36.429:

src NBMA: 172.21.100.1


*Feb 1 01:31:36.429:

src protocol: 10.10.10.1, dst protocol: 10.10.10.2


*Feb 1 01:31:36.429: (C-1) code: no error(0), flags: none
*Feb 1 01:31:36.429: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:36.429: addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 0, pref: 255
*Feb 1 01:31:36.429: Responder Address Extension(3):
*Feb 1 01:31:36.429: Forward Transit NHS Record Extension(4):
```

*Feb 1 01:31:36.429: Reverse Transit NHS Record Extension(5):
*Feb 1 01:31:36.429: Authentication Extension(7):
*Feb 1 01:31:36.429: type:Cleartext(1),

data:DMVPN

*Feb 1 01:31:36.429: NAT address Extension(9):
*Feb 1 01:31:36.430: NHRP: Encapsulation succeeded. Sending NHRP Control Packet NBMA Address: 172.20.10
*Feb 1 01:31:36.430: NHRP: 109 bytes out Tunnel10
*Feb 1 01:31:36.430: NHRP-RATE:

Retransmitting Resolution Request for 10.10.10.2, reqid 10, (retrans ivl 4 sec)

*Feb 1 01:31:39.816: NHRP: Checking for delayed event NULL/10.10.10.2 on list (Tunnel10 vrf: global(0x0)
*Feb 1 01:31:39.816: NHRP: No delayed event node found.
*Feb 1 01:31:39.816: NHRP: There is no VPE Extension to construct for the request
*Feb 1 01:31:39.817: NHRP: Sending NHRP Resolution Request for dest: 10.10.10.2 to nexthop: 10.10.10.2
*Feb 1 01:31:39.817: NHRP: Attempting to send packet through interface Tunnel10 via DEST dst 10.10.10.2
*Feb 1 01:31:39.817: NHRP-DETAIL: First hop route lookup for 10.10.10.2 yielded 10.10.10.2, Tunnel10
*Feb 1 01:31:39.817: NHRP:

Send Resolution Request via Tunnel10 vrf: global(0x0), packet size: 85

*Feb 1 01:31:39.817: src: 10.10.10.1, dst: 10.10.10.2
*Feb 1 01:31:39.817: (F) afn: AF_IP(1), type: IP(800), hop: 255, ver: 1
*Feb 1 01:31:39.817: shtl: 4(NSAP), sstl: 0(NSAP)
*Feb 1 01:31:39.817: pktsz: 85 extoff: 52
*Feb 1 01:31:39.817: (M) flags: "router auth src-stable nat ",

reqid: 10

*Feb 1 01:31:39.817:

src NBMA: 172.21.100.1

*Feb 1 01:31:39.817:

src protocol: 10.10.10.1, dst protocol: 10.10.10.2

*Feb 1 01:31:39.817: (C-1) code: no error(0), flags: none
*Feb 1 01:31:39.817: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:39.817: addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 0, pref: 255
*Feb 1 01:31:39.817: Responder Address Extension(3):
*Feb 1 01:31:39.817: Forward Transit NHS Record Extension(4):
*Feb 1 01:31:39.817: Reverse Transit NHS Record Extension(5):
*Feb 1 01:31:39.817: Authentication Extension(7):
*Feb 1 01:31:39.817: type:Cleartext(1),

data:DMVPN

*Feb 1 01:31:39.817: NAT address Extension(9):
*Feb 1 01:31:39.817: NHRP: Encapsulation succeeded. Sending NHRP Control Packet NBMA Address: 172.20.10
*Feb 1 01:31:39.818: NHRP: 109 bytes out Tunnel10
*Feb 1 01:31:39.818: NHRP-RATE:

Retransmitting Resolution Request for 10.10.10.2, reqid 10, (retrans ivl 8 sec)

*Feb 1 01:31:46.039: NHRP: Checking for delayed event NULL/10.10.10.2 on list (Tunnel10 vrf: global(0x0)

```
*Feb 1 01:31:46.040: NHRP: No delayed event node found.
*Feb 1 01:31:46.040: NHRP: There is no VPE Extension to construct for the request
```

Una vez que comienza el proceso NHRP de Spoke1, los registros muestran que el dispositivo está enviando la solicitud de resolución NHRP. El paquete tiene información importante como src NMBA y src protocol que son la dirección IP de NBMA y la dirección IP de túnel del spoke de origen (Spoke1). También puede ver el valor del protocolo dst que tiene la dirección IP del túnel del spoke de destino (Spoke2). Esto indica que Spoke1 solicita la dirección NBMA de Spoke2 para completar la asignación. También en el paquete, puede encontrar el valor required que puede ayudarlo a rastrear el paquete a lo largo de la trayectoria. Este valor seguirá siendo el mismo a lo largo de todo el proceso y puede ser útil para realizar un seguimiento de un flujo específico de la negociación NHRP. El paquete tiene otros valores que son importantes para la negociación, como la cadena de autenticación NHRP.

Una vez que el dispositivo envía la solicitud de resolución NHRP, los registros muestran que se envía una retransmisión. Esto se debe a que el dispositivo no ve la respuesta de resolución NHRP, por lo que envía el paquete nuevamente. Dado que Spoke1 no ve la respuesta, es necesario realizar un seguimiento de ese paquete en el siguiente dispositivo de la ruta, lo que significa el HUB.

Resultado de depuración del HUB:

<#root>

```
*Feb 1 01:31:34.262:
NHRP: Receive Resolution Request via Tunnel10 vrf: global(0x0), packet size: 85

*Feb 1 01:31:34.262: (F) afn: AF_IP(1), type: IP(800), hop: 255, ver: 1
*Feb 1 01:31:34.262: shtl: 4(NSAP), sstl: 0(NSAP)
*Feb 1 01:31:34.263: pktsz: 85 extoff: 52
*Feb 1 01:31:34.263: (M) flags: "router auth src-stable nat ",
reqid: 10

*Feb 1 01:31:34.263:
src NBMA: 172.21.100.1

*Feb 1 01:31:34.263:
src protocol: 10.10.10.1, dst protocol: 10.10.10.2

*Feb 1 01:31:34.263: (C-1) code: no error(0), flags: none
*Feb 1 01:31:34.263: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:34.263: addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 0, pref: 255
*Feb 1 01:31:34.263: Responder Address Extension(3):
*Feb 1 01:31:34.263: Forward Transit NHS Record Extension(4):
*Feb 1 01:31:34.263: Reverse Transit NHS Record Extension(5):
*Feb 1 01:31:34.263: Authentication Extension(7):
*Feb 1 01:31:34.263: type:Cleartext(1), data:DMVPN
```

```
*Feb 1 01:31:34.263: NAT address Extension(9):
*Feb 1 01:31:34.263: NHRP-DETAIL: netid_in = 10, to_us = 0
*Feb 1 01:31:34.263: NHRP-DETAIL:
```

Resolution request for afn 1 received on interface Tunnel10

```
 , for vrf: global(0x0) label: 0
*Feb 1 01:31:34.263: NHRP-DETAIL: Multipath IP route lookup for 10.10.10.2 in vrf: global(0x0) yielded
*Feb 1 01:31:34.263: NHRP:
```

Route lookup for destination 10.10.10.2

```
 in vrf: global(0x0) yielded interface Tunnel10, prefixlen 24
*Feb 1 01:31:34.263: NHRP-DETAIL: netid_out 10, netid_in 10
*Feb 1 01:31:34.263: NHRP: Forwarding request due to authoritative request.
*Feb 1 01:31:34.263: NHRP-ATTR:
```

NHRP Resolution Request packet is forwarded to 10.10.10.2 using vrf: global(0x0)

```
*Feb 1 01:31:34.263: NHRP: Attempting to forward to destination: 10.10.10.2 vrf: global(0x0)
*Feb 1 01:31:34.264: NHRP: Forwarding: NHRP SAS picked source: 10.10.10.10 for destination: 10.10.10.2
*Feb 1 01:31:34.264: NHRP: Attempting to send packet through interface Tunnel10 via DEST dst 10.10.10.2
*Feb 1 01:31:34.264: NHRP-DETAIL: First hop route lookup for 10.10.10.2 yielded 10.10.10.2, Tunnel10
*Feb 1 01:31:34.264: NHRP:
```

Forwarding Resolution Request via Tunnel10 vrf: global(0x0), packet size: 105

```
*Feb 1 01:31:34.264: src: 10.10.10.10, dst: 10.10.10.2
*Feb 1 01:31:34.264: (F) afn: AF_IP(1), type: IP(800), hop: 254, ver: 1
*Feb 1 01:31:34.264: shtl: 4(NSAP), sstl: 0(NSAP)
*Feb 1 01:31:34.264: pktsz: 105 extoff: 52
*Feb 1 01:31:34.264: (M) flags: "router auth src-stable nat ",
```

reqid: 10

```
*Feb 1 01:31:34.264:
```

src NBMA: 172.21.100.1

```
*Feb 1 01:31:34.264:
```

src protocol: 10.10.10.1, dst protocol: 10.10.10.2

```
*Feb 1 01:31:34.264: (C-1) code: no error(0), flags: none
*Feb 1 01:31:34.264: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:34.264: addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 0, pref: 255
*Feb 1 01:31:34.264: Responder Address Extension(3):
*Feb 1 01:31:34.264: Forward Transit NHS Record Extension(4):
*Feb 1 01:31:34.264: (C-1)
```

code: no error(0)

```
, flags: none
*Feb 1 01:31:34.264: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:34.264: addr_len: 4(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 255
*Feb 1 01:31:34.264:
```

client NBMA: 172.20.10.10

```
*Feb 1 01:31:34.264:
```

```
client protocol: 10.10.10.10


*Feb 1 01:31:34.264: Reverse Transit NHS Record Extension(5):
*Feb 1 01:31:34.264: Authentication Extension(7):
*Feb 1 01:31:34.264: type:Cleartext(1),

data:DMVPN


*Feb 1 01:31:34.265: NAT address Extension(9):
*Feb 1 01:31:34.265: NHRP: Encapsulation succeeded. Sending NHRP Control Packet NBMA Address: 172.22.20
*Feb 1 01:31:34.265: NHRP: 129 bytes out Tunnel10
```

Utilizando el valor de required, puede observar que el HUB recibe la solicitud de resolución enviada por Spoke1. En el paquete, los valores de src NBMA y src protocol son la información de Spoke1, y el valor de dst protocol es la IP de túnel de Spoke2, como se vio en las depuraciones de Spoke1. Cuando el HUB recibe la solicitud de resolución, realiza una búsqueda de ruta y reenvía el paquete a Spoke2. En el paquete reenviado, el HUB agrega una extensión que contiene su propia información (dirección IP de NBMA y dirección IP del túnel).

Las depuraciones anteriores muestran que el HUB está reenviando correctamente la solicitud de resolución a spoke 2. Por lo tanto, el siguiente paso es confirmar que Spoke2 lo recibe, lo procesa correctamente y envía a Spoke1 la respuesta de resolución.

Resultado de depuración de Spoke2:

<#root>

```
---------- [IKE/IPSEC DEBUG OUTPUTS OMITTED]----------

*Feb 1 01:31:34.647: ISAKMP: (1015):

Old State = IKE_QM_IPSEC_INSTALL_AWAIT New State = IKE_QM_PHASE2_COMPLETE


*Feb 1 01:31:34.647: NHRP: Process delayed resolution request src:10.10.10.1 dst:10.10.10.2 vrf: global
*Feb 1 01:31:34.648: NHRP-DETAIL: Resolution request for afn 1 received on interface Tunnel10 , for vrf
*Feb 1 01:31:34.648: NHRP-DETAIL: Multipath IP route lookup for 10.10.10.2 in vrf: global(0x0) yielded
*Feb 1 01:31:34.648: NHRP:

Route lookup for destination 10.10.10.2 in vrf: global(0x0) yielded interface Tunnel10, prefixlen 24


*Feb 1 01:31:34.648: NHRP-ATTR: smart spoke feature and attributes are not configured
*Feb 1 01:31:34.648:

NHRP:


Request was to us. Process the NHRP Resolution Request.


*Feb 1 01:31:34.648: NHRP-DETAIL: Multipath IP route lookup for 10.10.10.2 in vrf: global(0x0) yielded
*Feb 1 01:31:34.648: NHRP: nhrp_rtlookup for 10.10.10.2 in vrf: global(0x0) yielded interface Tunnel10,
*Feb 1 01:31:34.648: NHRP: Request was to us, responding with ouraddress
```

```
*Feb 1 01:31:34.648: NHRP: Checking for delayed event 10.10.10.1/10.10.10.2 on list (Tunnel10 vrf: globa
*Feb 1 01:31:34.648: NHRP: No delayed event node found.
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10: Checking to see if we need to delay for src 172.22.200.2 dst 1
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10: crypto_ss_listen_start already listening
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Opening a socket with profile IPSEC
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F10
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Socket is already open. Ignoring.
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F10
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): tunnel is already open!
*Feb 1 01:31:34.648: NHRP: No need to delay processing of resolution event NBMA src:172.22.200.2 NBMA ds
*Feb 1 01:31:34.648: NHRP-MEF: No vendor private extension in NHRP packet
*Feb 1 01:31:34.649: NHRP-CACHE: Tunnel10: Cache update for target 10.10.10.1/32 vrf: global(0x0) label
*Feb 1 01:31:34.649: 172.21.100.1 (flags:0x2080)
*Feb 1 01:31:34.649: NHRP:
```

Adding Tunnel Endpoints (VPN: 10.10.10.1, NBMA: 172.21.100.1)

```
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10: crypto_ss_listen_start already listening
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Opening a socket with profile IPSEC
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F10
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Found an existing tunnel endpoint
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): tunnel_protection_stop_pending_time
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Socket is already open. Ignoring.
*Feb 1 01:31:34.653:
```

NHRP: Successfully attached NHRP subblock for Tunnel Endpoints (VPN: 10.10.10.1, NBMA: 172.21.100.1)

```
*Feb 1 01:31:34.653: NHRP: Peer capability:0
*Feb 1 01:31:34.653: NHRP-CACHE: Inserted subblock node(1 now) for cache: Target 10.10.10.1/32 nhop 10.1
*Feb 1 01:31:34.653: NHRP-CACHE: Converted internal dynamic cache entry for 10.10.10.1/32 interface Tunn
*Feb 1 01:31:34.653: NHRP-EVE: NHP-UP: 10.10.10.1, NBMA: 172.21.100.1
*Feb 1 01:31:34.653: NHRP-MEF: No vendor private extension in NHRP packet
*Feb 1 01:31:34.653: NHRP-CACHE: Tunnel10: Internal Cache add for target 10.10.10.2/32 vrf: global(0x0)
*Feb 1 01:31:34.653: 172.22.200.2 (flags:0x20)
*Feb 1 01:31:34.653: NHRP: Attempting to send packet through interface Tunnel10 via DEST dst 10.10.10.1
*Feb 1 01:31:34.654: NHRP-DETAIL: First hop route lookup for 10.10.10.1 yielded 10.10.10.1, Tunnel10
*Feb 1 01:31:34.654:
```

NHRP: Send Resolution Reply via Tunnel10 vrf: global(0x0), packet size: 133

```
*Feb 1 01:31:34.654: src: 10.10.10.2, dst: 10.10.10.1
*Feb 1 01:31:34.654: (F) afn: AF_IP(1), type: IP(800), hop: 255, ver: 1
*Feb 1 01:31:34.654: shtl: 4(NSAP), sstl: 0(NSAP)
*Feb 1 01:31:34.654: pktsz: 133 extoff: 60
*Feb 1 01:31:34.654: (M) flags: "router auth dst-stable unique src-stable nat ",
```

reqid: 10

```
*Feb 1 01:31:34.654:
```

src NBMA: 172.21.100.1

```
*Feb 1 01:31:34.654:
```

src protocol: 10.10.10.1, dst protocol: 10.10.10.2

```
*Feb 1 01:31:34.654: (C-1) code: no error(0), flags: none
*Feb 1 01:31:34.654: prefix: 32, mtu: 9976, hd_time: 599
*Feb 1 01:31:34.654: addr_len: 4(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 255
```

```
*Feb 1 01:31:34.654:
client NBMA: 172.22.200.2

*Feb 1 01:31:34.654:
client protocol: 10.10.10.2

*Feb 1 01:31:34.654: Responder Address Extension(3):
*Feb 1 01:31:34.654: (C) code: no error(0), flags: none
*Feb 1 01:31:34.654: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:34.654: addr_len: 4(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 255
*Feb 1 01:31:34.654:
client NBMA: 172.22.200.2

*Feb 1 01:31:34.654:
client protocol: 10.10.10.2

*Feb 1 01:31:34.654: Forward Transit NHS Record Extension(4):
*Feb 1 01:31:34.654: (C-1) code: no error(0), flags: none
*Feb 1 01:31:34.654: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:34.654: addr_len: 4(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 255
*Feb 1 01:31:34.654:
client NBMA: 172.20.10.10

*Feb 1 01:31:34.654:
client protocol: 10.10.10.10

*Feb 1 01:31:34.654: Reverse Transit NHS Record Extension(5):
*Feb 1 01:31:34.654: Authentication Extension(7):
*Feb 1 01:31:34.654: type:Cleartext(1),
data:DMVPN

*Feb 1 01:31:34.655: NAT address Extension(9):
*Feb 1 01:31:34.655: NHRP: Encapsulation succeeded. Sending NHRP Control Packet NBMA Address: 172.21.10
*Feb 1 01:31:34.655: NHRP: 157 bytes out Tunnel10
*Feb 1 01:31:34.655: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F1
*Feb 1 01:31:34.655: NHRP-DETAIL: Deleted delayed event on interfaceTunnel10 dest: 172.21.100.1
```

El reqid coincide con el valor visto en las salidas anteriores, con esto, se confirma que el paquete de solicitud de resolución NHRP enviado por Spoke1 alcanza Spoke2. Este paquete activa una búsqueda de ruta en Spoke2 y se da cuenta de que la solicitud de resolución es para sí misma, por lo tanto, Spoke2 agrega la información de Spoke1 a su tabla NHRP. Antes de enviar el paquete de respuesta de resolución a Spoke1, el dispositivo agrega su propia información (dirección IP de NBMA y dirección IP del túnel) para que Spoke1 pueda utilizar ese paquete para agregar esa información a su base de datos.

Según todas las depuraciones vistas, la respuesta de resolución NHRP enviada desde Spoke2 no

llega a Spoke1. El HUB se puede descartar del problema ya que está recibiendo y reenviando el paquete de solicitud de resolución NHRP como se esperaba. Por lo tanto, el siguiente paso es realizar capturas entre Spoke1 y Spoke2 para obtener más detalles sobre el problema.

Captura de paquetes integrada

La función de captura de paquetes integrada le permite analizar el tráfico que pasa a través del dispositivo. El primer paso para configurarlo es crear una lista de acceso que incluya el tráfico que desea capturar en ambos flujos de tráfico (entrante y saliente).

Para este escenario, se utilizan las direcciones IP de NBMA:

```
ip access-list extended filter
10 permit ip host 172.21.100.1 host 172.22.200.2
20 permit ip host 172.22.200.2 host 172.21.100.1
```

A continuación, configure la captura mediante el comando monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 interface <WAN_INTERFACE> both e inicie la captura con el comando monitor capture <CAPTURE_NAME>start.

Capturar configuración en Spoke1 y Spoke2:

```
monitor capture CAP access-list filter buffer size 10 interface GigabitEthernet1 both
monitor capture CAP start
```

Para mostrar el resultado de la captura, utilice el comando show monitor capture <CAPTURE_NAME> buffer brief.

Salida de captura Spoke1:

<#root>

```
SPOKE1#show monitor capture CAP buffer brief
------------------------------------------------------------------------------
#   size   timestamp     source             destination      dscp    protocol
------------------------------------------------------------------------------
   0  210    0.000000   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   1  150    0.014999   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   2  478    0.028990   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   3  498    0.049985   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   4  150    0.069988   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   5  134    0.072994   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   6  230    0.074993   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   7  230    0.089992   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   8  118    0.100993   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   9  218    0.108988   172.22.200.2    ->  172.21.100.1     48 CS6  ESP
```

```
  10   70    0.108988   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  11  218    1.907994   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  12   70    1.907994   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  13  218    5.818003   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  14   70    5.818003   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  15  218   12.559969   172.22.200.2    ->  172.21.100.1    48 CS6  ESP

  16   70   12.559969   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  17  218   26.859001   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  18   70   26.859001   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  19  218   54.378978   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  20   70   54.378978   172.21.100.1    ->  172.22.200.2     0  BE   ICMP
```

Salida de captura Spoke2:

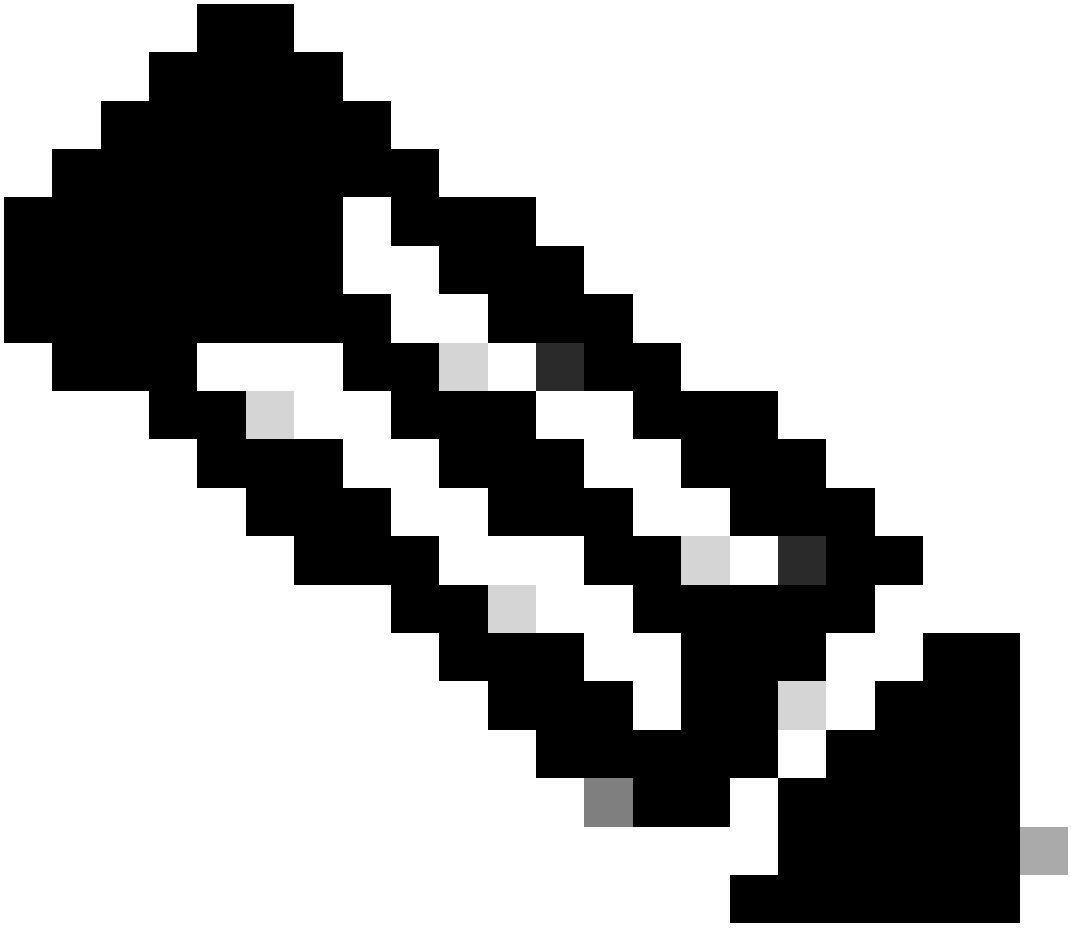
<#root>

```
SPOKE2#show monitor capture CAP buffer brief
------------------------------------------------------------------------
#   size   timestamp     source             destination      dscp    protocol
------------------------------------------------------------------------
   0  210    0.000000   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   1  150    0.015990   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   2  478    0.027998   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   3  498    0.050992   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   4  150    0.069988   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   5  134    0.072994   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   6  230    0.074993   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
   7  230    0.089992   172.21.100.1    ->  172.22.200.2     48 CS6  UDP
   8  118    0.099986   172.22.200.2    ->  172.21.100.1     48 CS6  UDP
```

```
9  218    0.108988   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  10   70    0.108988   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  11  218    1.907994   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  12   70    1.909001   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  13  218    5.817011   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  14   70    5.818002   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  15  218   12.559968   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  16   70   12.560960   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  17  218   26.858009   172.22.200.2    ->  172.21.100.1     48 CS6  ESP

  18   70   26.859001   172.21.100.1    ->  172.22.200.2     0  BE   ICMP

  19  218   54.378978   172.22.200.2   ->  172.21.100.1     48 CS6  ESP

  20   70   54.379970   172.21.100.1    ->  172.22.200.2     0  BE   ICMP
```

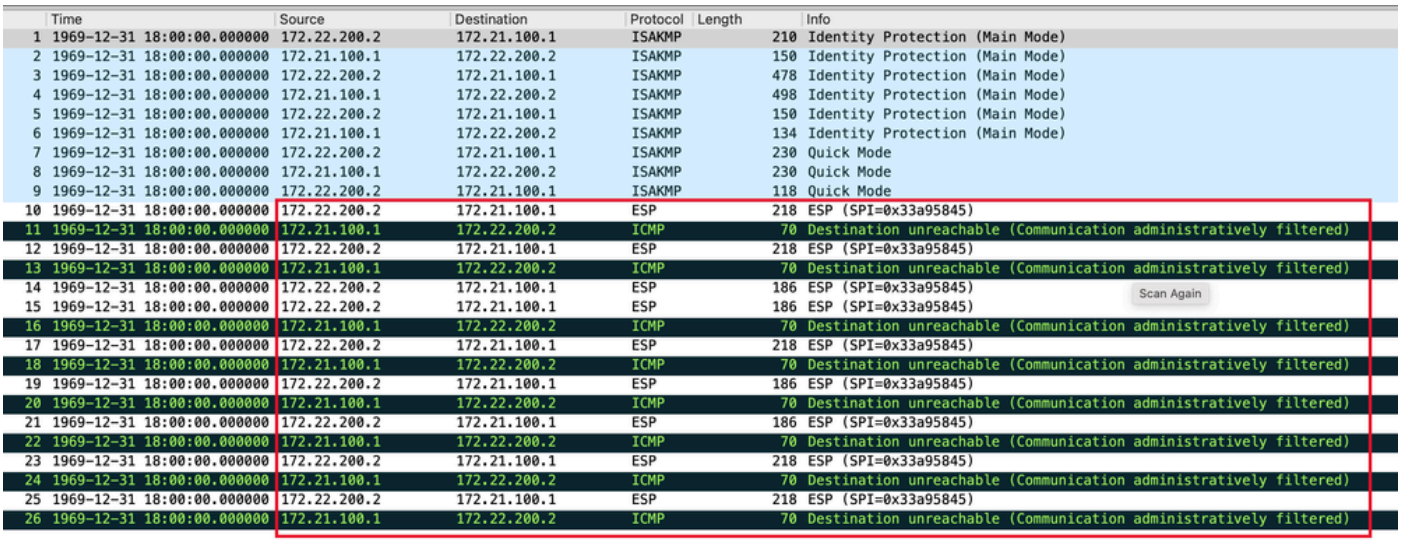
El resultado de las capturas muestra que los paquetes iniciales son tráfico UDP, lo que indica la negociación IKE/IPSEC. Después de eso, Spoke2 envía la respuesta de resolución a Spoke1, que se ve como tráfico ESP (paquete 9). Después de esto, el flujo de tráfico esperado es ESP, sin embargo, el siguiente paquete visto es el tráfico ICMP que viene de Spoke1 a Spoke2.

Para analizar más a fondo el paquete, puede exportar el archivo pcap desde el dispositivo ejecutando el comando show monitor capture <CAPTURE_NAME> buffer dump. A continuación, utilice una herramienta de descodificación para convertir el resultado del volcado en un archivo pcap y poder abrirlo con Wireshark.



Nota: Cisco cuenta con un analizador de paquetes donde puede encontrar configuraciones de captura, ejemplos y un descodificador: Herramienta TAC de Cisco - Generador y analizador de configuración de captura de paquetes

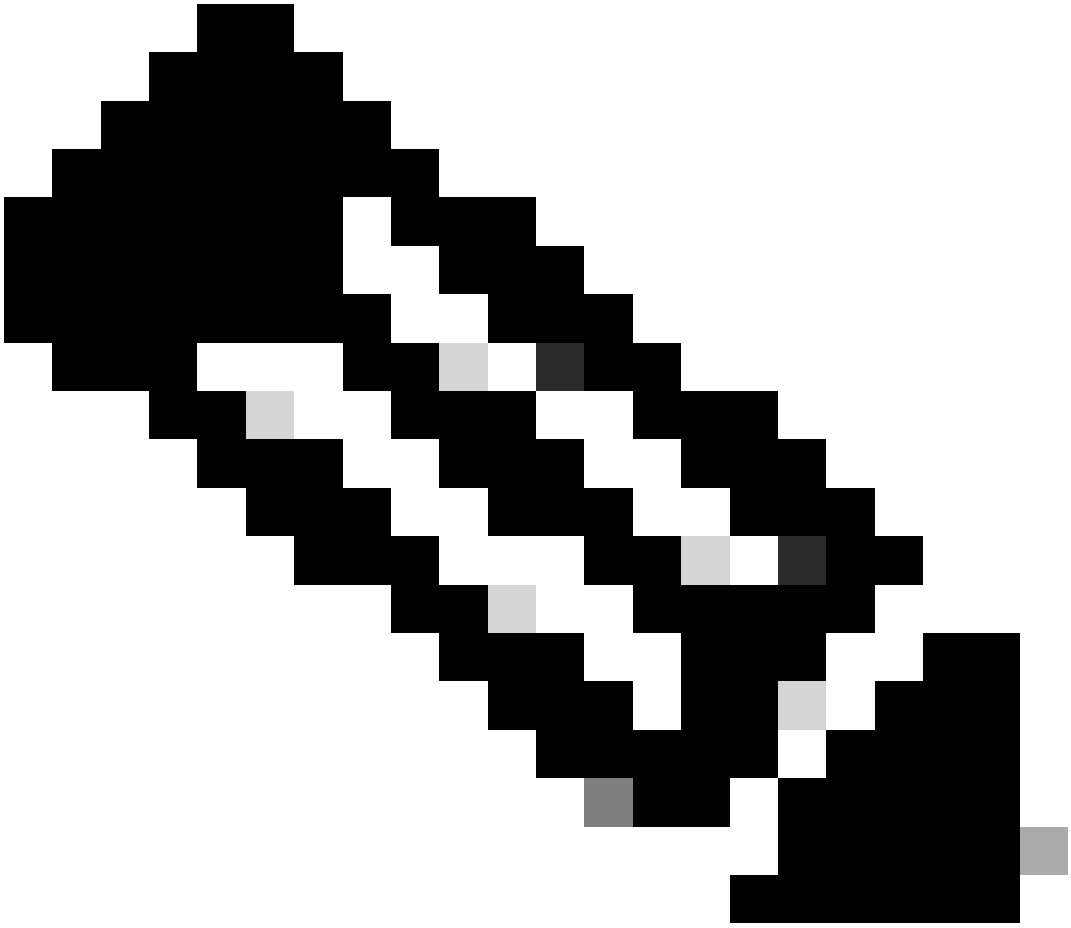
Salida de Wireshark:



	Time	Source	Destination	Protocol	Length	Info
1	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	210	Identity Protection (Main Mode)
2	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	150	Identity Protection (Main Mode)
3	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	478	Identity Protection (Main Mode)
4	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	498	Identity Protection (Main Mode)
5	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	150	Identity Protection (Main Mode)
6	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	134	Identity Protection (Main Mode)
7	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	230	Quick Mode
8	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	230	Quick Mode
9	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	118	Quick Mode
10	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	218	ESP (SPI=0x33a95845)
11	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
12	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	218	ESP (SPI=0x33a95845)
13	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
14	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	186	ESP (SPI=0x33a95845)
15	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	186	ESP (SPI=0x33a95845)
16	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
17	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	218	ESP (SPI=0x33a95845)
18	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
19	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	186	ESP (SPI=0x33a95845)
20	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
21	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	186	ESP (SPI=0x33a95845)
22	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
23	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	218	ESP (SPI=0x33a95845)
24	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)
25	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ESP	218	ESP (SPI=0x33a95845)
26	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ICMP	70	Destination unreachable (Communication administratively filtered)

Capturar salida en Wireshark

El contenido del paquete ICMP tiene el mensaje de error Destination unreachable (Communication administrativamente filtrado). Esto indica que existe algún tipo de filtro, como una ACL de router o un firewall que afecta el tráfico a lo largo de la trayectoria. La mayoría de las veces, el filtro se configura en el dispositivo que envía el paquete (en este caso, Spoke1), pero los dispositivos intermedios también pueden enviarlo.



Nota: La salida de Wireshark es la misma en ambos radios.

Función Cisco IOS® XE Datapath Packet Trace

La función de seguimiento de paquetes de ruta de datos de Cisco IOS XE se utiliza para analizar cómo el dispositivo está procesando el tráfico. Para configurarlo, debe crear una lista de acceso que incluya el tráfico que desea capturar en ambos flujos de tráfico (entrante y saliente).

Para este escenario, se utilizan las direcciones IP de NBMA.

```
ip access-list extended filter
10 permit ip host 172.21.100.1 host 172.22.200.2
20 permit ip host 172.22.200.2 host 172.21.100.1
```

Luego, configure la función fia-trace y establezca la condición de depuración para utilizar la lista

de acceso. Finalmente, comienza la condición.

```
debug platform packet-trace packet 1024 fia-trace
debug platform condition ipv4 access-list filter both
debug platform condition start
```

- debug platform packet-trace packet <count> fia-trace: habilita el seguimiento detallado de fia y lo detiene una vez que se ha capturado la cantidad de paquetes configurados
- debug platform condition ipv4 access-list <ACL-NAME> both: establece una condición en el dispositivo mediante la lista de acceso previamente configurada
- debug platform condition start: inicia la condición

Para revisar el resultado de fia-trace, utilice los siguientes comandos.

```
show platform packet-trace statistics
show platform packet-trace summary
show platform packet-trace packet <number>
```

Spoke1 show platform packet-trace statistics output:

<#root>

```
SPOKE1#show platform packet-trace statistics
Packets Summary
  Matched  18
  Traced   18
Packets Received
  Ingress  11
  Inject   7
    Count       Code  Cause
    4           2     QFP destination lookup
    3           9     QFP ICMP generated packet
Packets Processed
  Forward  7
  Punt     8
    Count       Code  Cause
    5           11    For-us data
    3           26    QFP ICMP generated packet
  Drop     3

    Count       Code  Cause

    3           8     Ipv4Acl

  Consume  0
```

```
          PKT_DIR_IN
             Dropped       Consumed       Forwarded
INFRA            0              0              0
TCP              0              0              0
UDP              0              0              5
IP               0              0              5
IPV6             0              0              0
ARP              0              0              0

          PKT_DIR_OUT
             Dropped       Consumed       Forwarded
INFRA            0              0              0
TCP              0              0              0
UDP              0              0              0
IP               0              0              0
IPV6             0              0              0
ARP              0              0              0
```

En el resultado de show platform packet-trace statistics, puede ver los contadores de los paquetes procesados por el dispositivo. Esto le permite ver los paquetes entrantes y salientes, y verificar si el dispositivo está descartando paquetes, junto con la razón de la caída.

En el resultado que se muestra, Spoke1 está descartando algunos paquetes con la descripción Ipv4Acl. Para analizar más a fondo esos paquetes, se puede utilizar el comando show platform packet-trace summary.

Resultado del resumen de seguimiento de paquetes de Spoke1 show platform packet-trace:

<#root>

```
SPOKE1#show platform packet-trace summary
Pkt    Input                     Output                    State  Reason
0      Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
1      INJ.2                     Gi1                       FWD
2      Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
3      INJ.2                     Gi1                       FWD
4      Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
5      INJ.2                     Gi1                       FWD
6      Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
7      INJ.2                     Gi1                       FWD
8      Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)

9      Gi1                       Gi1                       DROP   8   (Ipv4Acl)


10     Gi1                       internal0/0/recycle:0     PUNT   26  (QFP ICMP generated packet)
11     INJ.9                     Gi1                       FWD

12     Gi1                       Gi1                       DROP   8   (Ipv4Acl)


13     Gi1                       internal0/0/recycle:0     PUNT   26  (QFP ICMP generated packet)
14     INJ.9                     Gi1                       FWD

15     Gi1                       Gi1                       DROP   8   (Ipv4Acl)
```

```
16    Gi1                     internal0/0/recycle:0     PUNT   26  (QFP ICMP generated packet)
17    INJ.9                   Gi1                       FWD
18    Gi1                     Gi1                       DROP   8   (Ipv4Acl)
19    Gi1                     internal0/0/recycle:0     PUNT   26  (QFP ICMP generated packet)
20    INJ.9                   Gi1                       FWD
21    Gi1                     Gi1                       DROP   8   (Ipv4Acl)
22    Gi1                     internal0/0/recycle:0     PUNT   26  (QFP ICMP generated packet)
23    INJ.9                   Gi1                       FWD
24    Gi1                     Gi1                       DROP   8   (Ipv4Acl)
25    Gi1                     internal0/0/recycle:0     PUNT   26  (QFP ICMP generated packet)
26    INJ.9                   Gi1                       FWD
```

Con esta salida, puede ver cada paquete que llega y sale del dispositivo, así como las interfaces de ingreso y egreso. También se muestra el estado del paquete, indicando si se reenvió, descartó o procesó internamente (punt).

En este ejemplo, este resultado ayudó a identificar los paquetes que el dispositivo está descartando. Con el comando show platform packet-trace packet <PACKET_NUMBER>, puede ver cómo el dispositivo procesa ese paquete específico.

Spoke1 show platform packet-trace packet <PACKET_NUMBER> output:

<#root>

```
SPOKE1#show platform packet-trace packet 9
Packet: 9 CBUG ID: 9
Summary
Input : GigabitEthernet1

Output : GigabitEthernet1

State : DROP 8 (Ipv4Acl)

Timestamp
Start : 366032715676920 ns (02/01/2024 04:30:15.708990 UTC)
Stop : 366032715714128 ns (02/01/2024 04:30:15.709027 UTC)
Path Trace
  Feature: IPV4(Input)

Input : GigabitEthernet1

     Output : <unknown>
```

Source : 172.22.200.2

Destination : 172.21.100.1

Protocol : 50 (ESP)

Feature: DEBUG_COND_INPUT_PKT
Entry : Input - 0x812707d0

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 194 ns
Feature: IPV4_INPUT_DST_LOOKUP_ISSUE
Entry : Input - 0x8129bf74

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 769 ns
Feature: IPV4_INPUT_ARL_SANITY
Entry : Input - 0x812725cc

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 307 ns
Feature: EPC_INGRESS_FEATURE_ENABLE
Entry : Input - 0x812782d0

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 6613 ns
Feature: IPV4_INPUT_DST_LOOKUP_CONSUME
Entry : Input - 0x8129bf70

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 272 ns
Feature: STILE_LEGACY_DROP
Entry : Input - 0x812a7650

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 278 ns
Feature: INGRESS_MMA_LOOKUP_DROP
Entry : Input - 0x812a1278

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 697 ns
Feature: INPUT_DROP_FNF_AOR
Entry : Input - 0x81297278

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 676 ns
Feature: INPUT_FNF_DROP
Entry : Input - 0x81280f24

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 1018 ns
Feature: INPUT_DROP_FNF_AOR_RELEASE
Entry : Input - 0x81297274

Input : GigabitEthernet1

Output : <unknown>

Lapsed time : 174 ns
Feature: INPUT_DROP

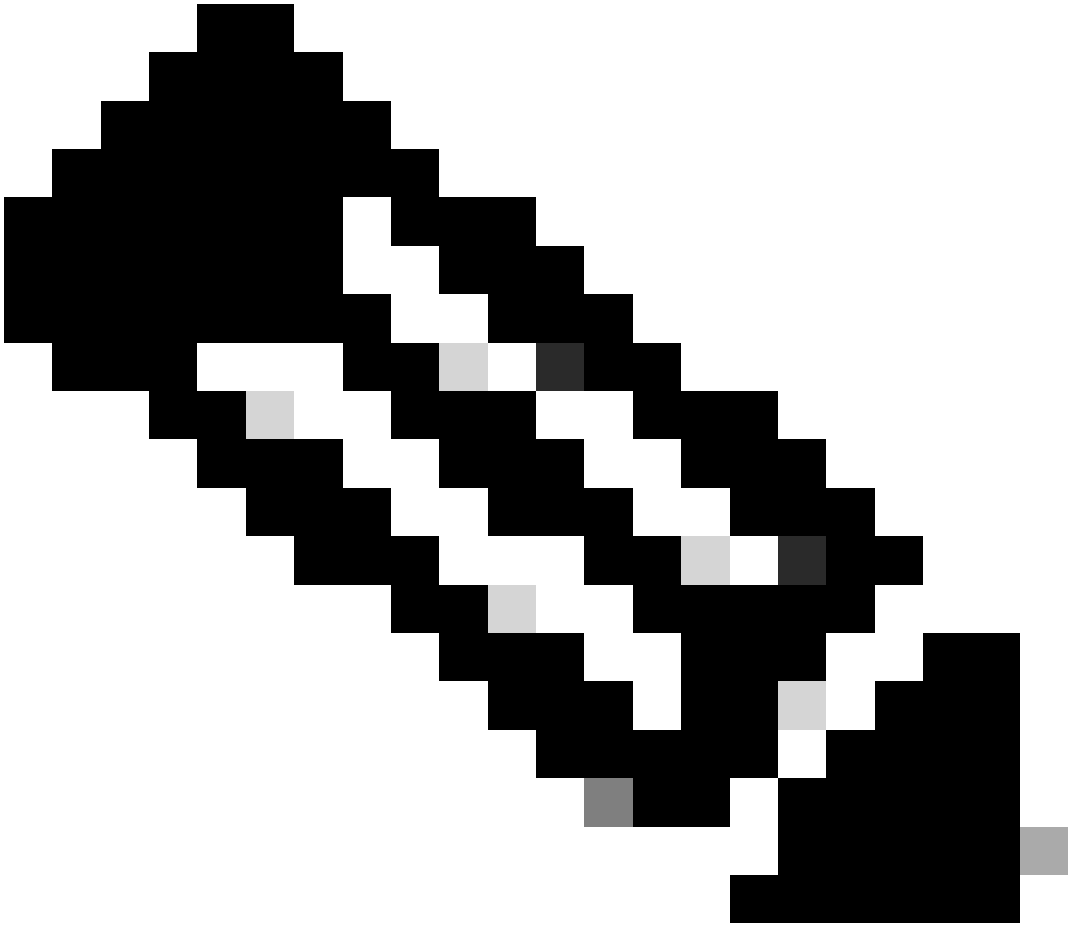
Entry : Input - 0x8126e568

```
Input : GigabitEthernet1

     Output : <unknown>

     Lapsed time : 116 ns

Feature: IPV4_INPUT_ACL

     Entry : Input - 0x81271f70

Input : GigabitEthernet1

     Output : <unknown>

     Lapsed time : 12915 ns
```

En la primera parte, puede ver la interfaz de ingreso y egreso, y el estado del paquete. Esto es seguido por la segunda parte del resultado donde puede encontrar las direcciones IP de origen y destino y el protocolo.

Cada fase posterior muestra cómo el dispositivo procesa este paquete en particular. Esto ofrece información sobre cualquier configuración, como la traducción de direcciones de red (NAT) o la lista de acceso u otros factores que podrían afectarla.

En este caso, se puede identificar que el protocolo del paquete es ESP, que la IP de origen es la dirección IP NBMA de Spoke2 y que la IP de destino es la dirección IP NBMA de Spoke1. Esto indica que este es el paquete faltante en la negociación NHRP. Además, se observa que no se especifica ninguna interfaz de salida en ninguna fase, lo que sugiere que algo afectó al tráfico antes de que se pudiera reenviar. En la penúltima fase puede ver que el dispositivo está descartando el tráfico entrante en la interfaz especificada (GigabitEthernet1). La última fase muestra una lista de acceso de entrada, lo que sugiere que puede haber alguna configuración en la interfaz que cause la caída.



Nota: Si después de utilizar todas las herramientas de troubleshooting enumeradas en este documento, los spokes involucrados en la negociación no muestran ninguna señal de que estén descartando o afectando el tráfico, entonces concluye la solución de problemas en esos dispositivos.

El siguiente paso debe ser comprobar los dispositivos intermedios entre ellos, como firewalls, switches e ISP.

Solución

Si se observa tal escenario, el siguiente paso es verificar la interfaz mostrada en los resultados anteriores. Esto implica verificar la configuración para verificar si hay algo que afecte el tráfico.

Configuración de la interfaz WAN:

<#root>

```
SPOKE1#show running-configuration interface gigabitEthernet1
Building configuration...

Current configuration : 150 bytes
!
interface GigabitEthernet1
ip address 172.21.100.1 255.255.255.0
ip access-group ESP_TRAFFIC in

negotiation auto
no mop enabled
no mop sysid
end
```

Como parte de su configuración, la interfaz tiene aplicado un grupo de acceso. Es importante verificar que los hosts configurados en la lista de acceso no interfieran con el tráfico utilizado para la negociación NHRP.

<#root>

```
SPOKE1#show access-lists ESP_TRAFFIC
Extended IP access list ESP_TRAFFIC
10 deny esp host 172.21.100.1 host 172.22.200.2
20 deny esp host 172.22.200.2 host 172.21.100.1 (114 matches)

30 permit ip any any (22748 matches)
```

La segunda sentencia de la lista de acceso niega la comunicación entre la dirección IP NBMA de Spoke2 y la dirección IP NBMA de Spoke1, causando la caída previamente vista. Después de quitar el grupo de acceso de la interfaz, la comunicación entre los dos spokes es exitosa:

```
SPOKE1#ping 192.168.2.2 source loopback1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/3 ms
```

El túnel IPSEC está activo y ahora muestra los encapsulados y los decaps en ambos dispositivos:

Spoke1:

<#root>

```
SPOKE1#show crypto IPSEC sa peer 172.22.200.2
```

```
interface: Tunnel10
    Crypto map tag: Tunnel10-head-0, local addr 172.21.100.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.21.100.1/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.22.200.2/255.255.255.255/47/0)
   current_peer 172.22.200.2 port 500
     PERMIT, flags={origin_is_acl,}


#pkts encaps: 6, #pkts encrypt: 6, #pkts digest: 6



    #pkts decaps: 7, #pkts decrypt: 7, #pkts verify: 7


    #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.: 172.21.100.1, remote crypto endpt.: 172.22.200.2
     plaintext mtu 1458, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
     current outbound spi: 0x9392DA81(2475874945)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0xBF8F523D(3213840957)
        transform: esp-256-aes esp-sha256-hmac ,
        in use settings ={Transport, }
        conn id: 2073, flow_id: CSR:73, sibling_flags FFFFFFFF80000008, crypto map: Tunnel10-head-0
         sa timing: remaining key lifetime (k/sec): (4607998/28783)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x9392DA81(2475874945)
        transform: esp-256-aes esp-sha256-hmac ,
        in use settings ={Transport, }
        conn id: 2074, flow_id: CSR:74, sibling_flags FFFFFFFF80000008, crypto map: Tunnel10-head-0
         sa timing: remaining key lifetime (k/sec): (4607999/28783)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:

     outbound pcp sas:
```

Spoke2

<#root>

```
SPOKE2#show crypto IPSEC sa peer 172.21.100.1

interface: Tunnel10
    Crypto map tag: Tunnel10-head-0, local addr 172.22.200.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.22.200.2/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.21.100.1/255.255.255.255/47/0)
   current_peer 172.21.100.1 port 500
     PERMIT, flags={origin_is_acl,}


#pkts encaps: 7, #pkts encrypt: 7, #pkts digest: 7



    #pkts decaps: 6, #pkts decrypt: 6, #pkts verify: 6


    #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.: 172.22.200.2, remote crypto endpt.: 172.21.100.1
     plaintext mtu 1458, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
     current outbound spi: 0xBF8F523D(3213840957)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0x9392DA81(2475874945)
        transform: esp-256-aes esp-sha256-hmac ,
        in use settings ={Transport, }
        conn id: 2073, flow_id: CSR:73, sibling_flags FFFFFFFF80000008, crypto map: Tunnel10-head-0
         sa timing: remaining key lifetime (k/sec): (4607998/28783)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xBF8F523D(3213840957)
        transform: esp-256-aes esp-sha256-hmac ,
        in use settings ={Transport, }
        conn id: 2074, flow_id: CSR:74, sibling_flags FFFFFFFF80000008, crypto map: Tunnel10-head-0
         sa timing: remaining key lifetime (k/sec): (4607999/28783)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:

     outbound pcp sas:
```

Ahora, la tabla DMVPN de Spoke1 muestra la asignación correcta en ambas entradas:

<#root>

```
SPOKE1#show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override, B - BGP
        C - CTS Capable, I2 - Temporary
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel10, IPv4 NHRP Details
Type:Spoke, NHRP Peers:2,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
----- --------------- --------------- ----- -------- -----


1 172.22.200.2     10.10.10.2    UP   00:01:31   D


     1 172.20.10.10     10.10.10.10    UP    1d05h     S
```

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).