Identificar e Solucionar Problemas do Túnel Spoke-to-Spoke da Fase 2 do DMVPN

Contents

Introdução

Este documento descreve como solucionar problemas de um túnel DMVPN spoke-to-spoke fase 2 quando ele não estabelece.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento sobre os próximos tópicos:

- Rede Virtual Privada Multiponto Dinâmica (DMVPN)
- protocolos IKE/IPSEC
- Protocolo de Resolução do Próximo Salto (NHRP)

Componentes Utilizados

Este documento é baseado nesta versão de software:

- Cisco CSR1000V (VXE) - Versão 17.03.08

As informações neste documento foram criadas a partir de dispositivos em um ambiente de

laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Este documento descreve como configurar e usar diferentes ferramentas de solução de problemas em um problema de DMVPN comum. O problema é a negociação com falha de um túnel DMVPN da fase 2, em que o spoke de origem, o estado DMVPN mostra UP com o mapeamento de túnel/multiacesso sem broadcast (NBMA) correto para o spoke de destino. No entanto, no spoke de destino, um mapeamento incorreto é exibido.

Contexto teórico

É importante entender como os túneis spoke-to-spoke são estabelecidos quando se tem uma configuração DMVPN Fase 2. Esta seção fornece um breve resumo teórico do processo NHRP durante esta fase.

Na fase 2 do DMVPN, você pode criar túneis spoke-to-spoke dinâmicos sob demanda. Isso é possível porque, em todos os dispositivos dentro da nuvem DMVPN (hub e spokes), o modo da interface de túnel muda para o multiponto Generic Routing Encapsulation (GRE). Um dos principais recursos dessa fase é que o hub não é percebido como o próximo salto pelos outros dispositivos. Em vez disso, todos os spokes têm as informações de roteamento um do outro. Ao estabelecer um túnel spoke-to-spoke na fase 2, um processo NHRP é acionado onde os spokes aprendem as informações sobre outros spokes e faz um mapeamento entre o NBMA e os endereços IP do túnel.

As próximas etapas listam como o processo de resolução do NHRP é acionado:

1. Quando o spoke de origem tenta acessar a LAN do spoke de destino, ele faz uma pesquisa de rota acionando a mensagem de solicitação de resolução para obter o endereço NBMA do spoke de destino. O spoke de origem envia essa mensagem inicial para o hub.

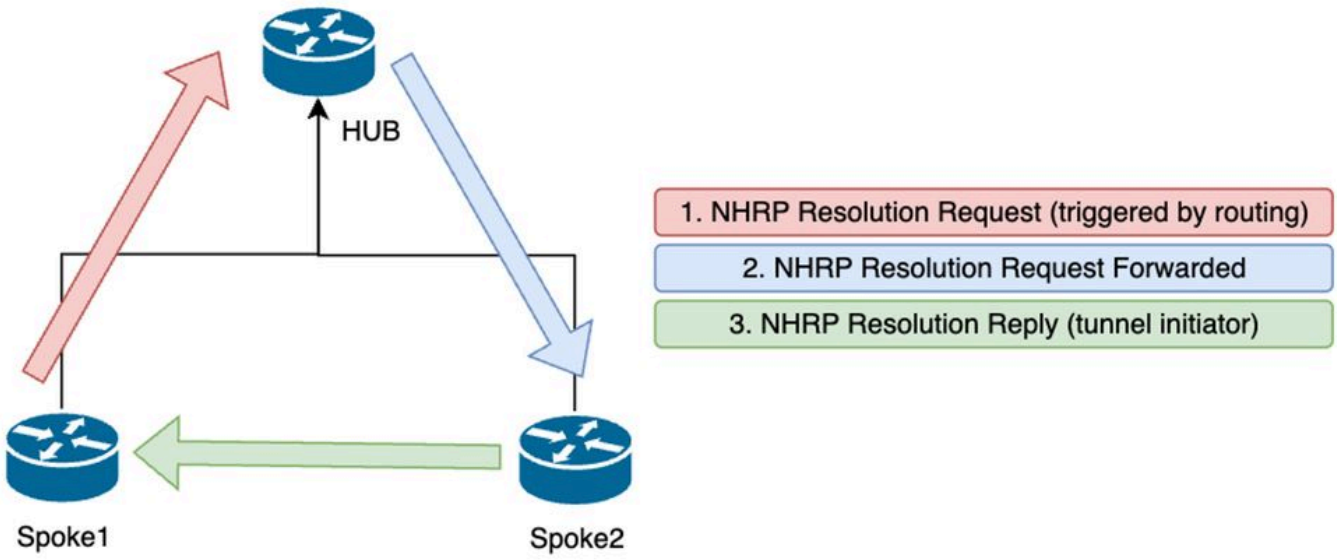
2. O hub recebe a solicitação de resolução e a encaminha para o spoke de destino.

3. O spoke de destino envia a resposta de resolução para o spoke de origem. Se a configuração do túnel tiver um perfil IPSEC vinculado:

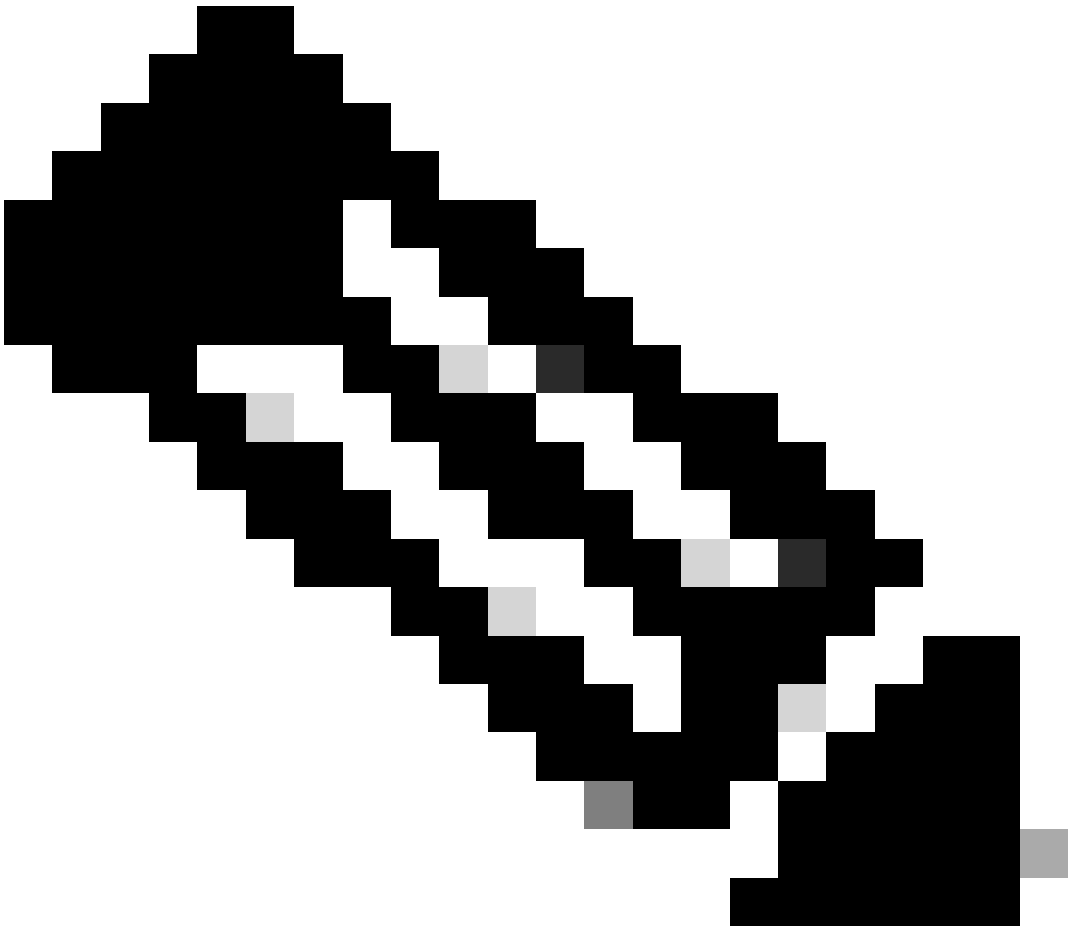
    - O processo de resolução do NHRP é atrasado até que os protocolos IKE/IPSEC possam ser estabelecidos.

    - O spoke de destino inicia e estabelece os túneis IKE/IPSEC.

    - Em seguida, o processo NHRP é retomado e o spoke de destino envia a resposta de resolução para o spoke de origem usando o túnel IPSEC como método de transporte.



Fluxo de mensagens NHRP entre os spokes na fase 2



Observação: antes de iniciar o processo de resolução, todos os spokes já devem estar

---

registrados no HUB.

Topologia

Este diagrama mostra a topologia usada para o cenário:

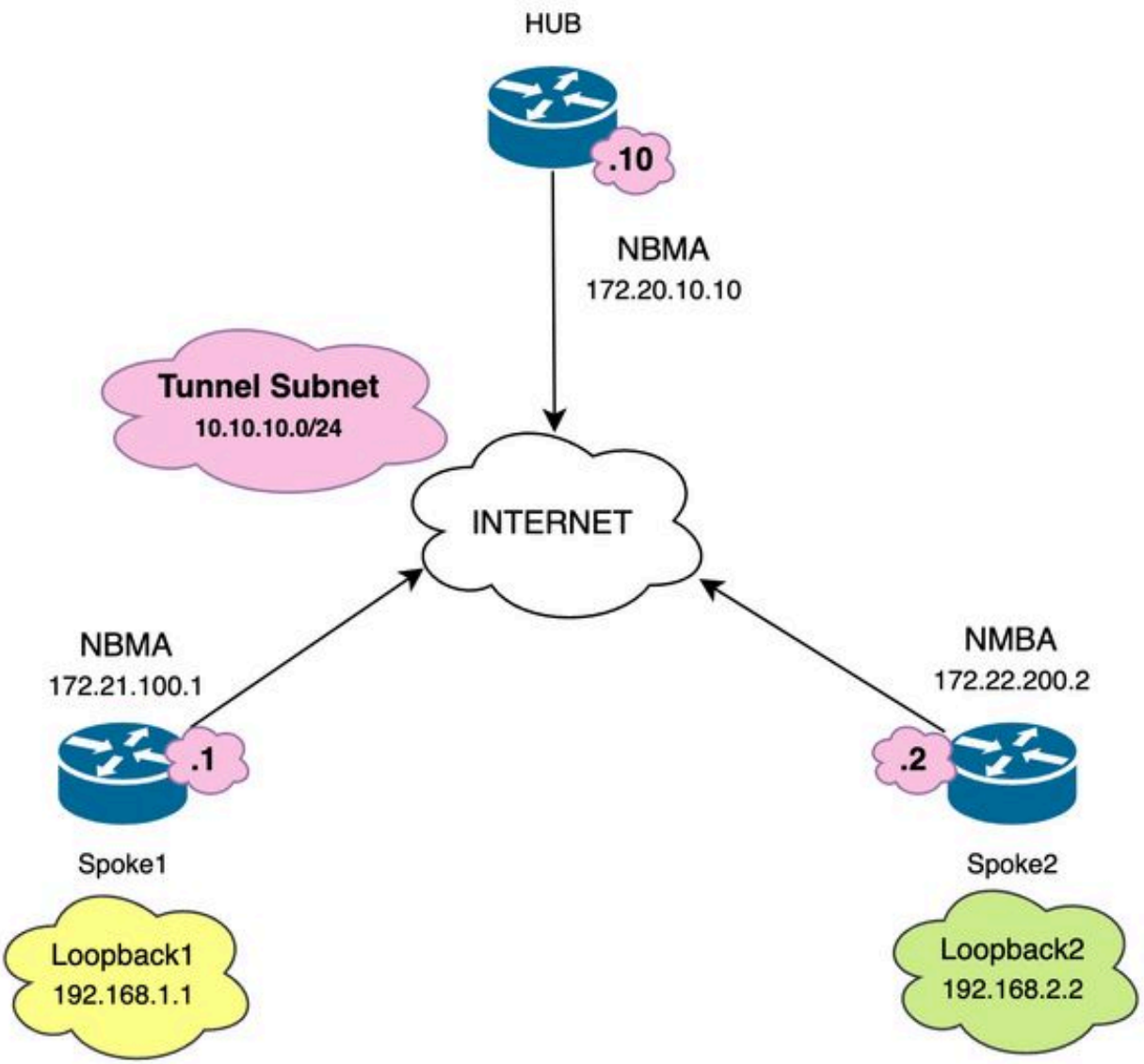


Diagrama de Rede e Sub-Redes IP Usadas

Passos de Troubleshooting

Nesse cenário, o túnel spoke-to-spoke entre Spoke1 e Spoke2 não é estabelecido, afetando a comunicação entre seus recursos locais (representados por interfaces de loopback), já que eles não são capazes de alcançar um ao outro.

```
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)
```

Validação inicial

Ao encontrar tal cenário, é importante começar validando a configuração do túnel e garantir que ambos os dispositivos tenham os valores corretos dentro dele. Para revisar a configuração do túnel, execute o comando show running-config interface tunnel<ID>.

Configuração do túnel do 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
```

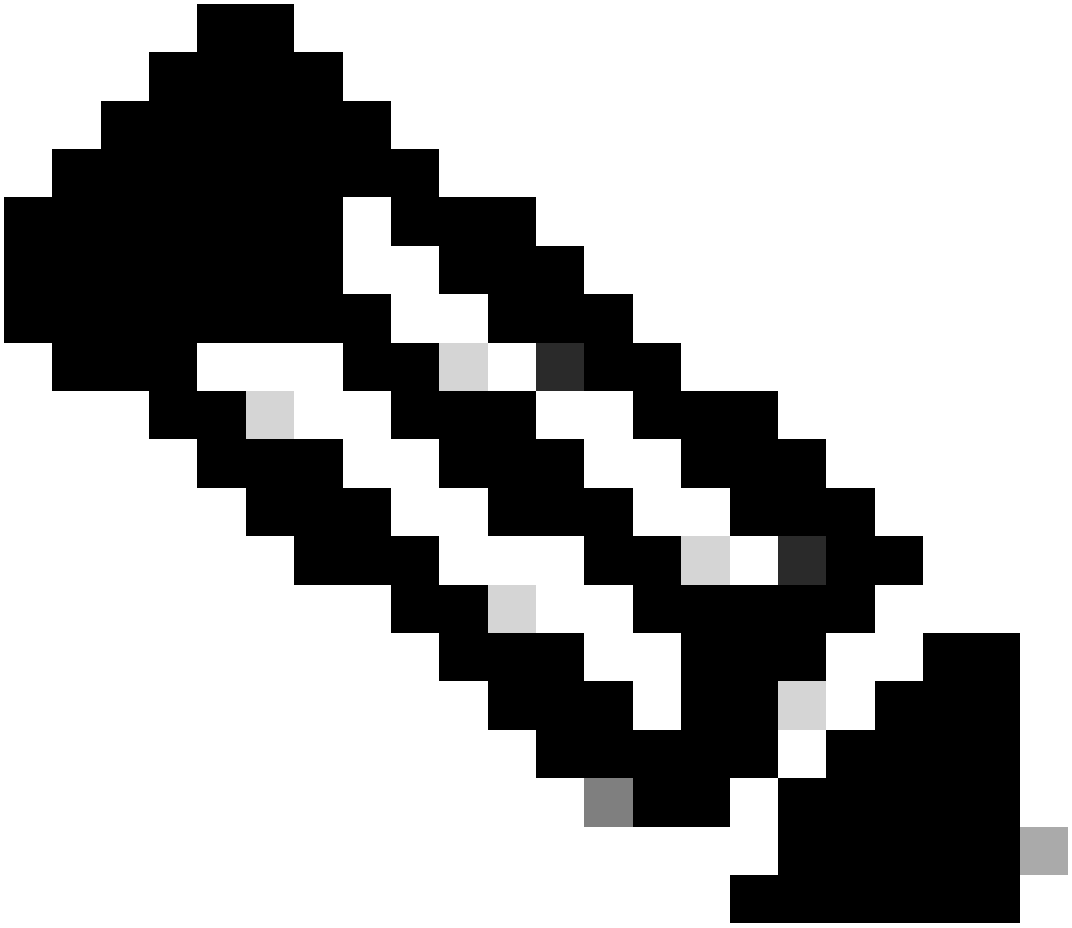
Configuração do túnel do spoke 2:

<#root>

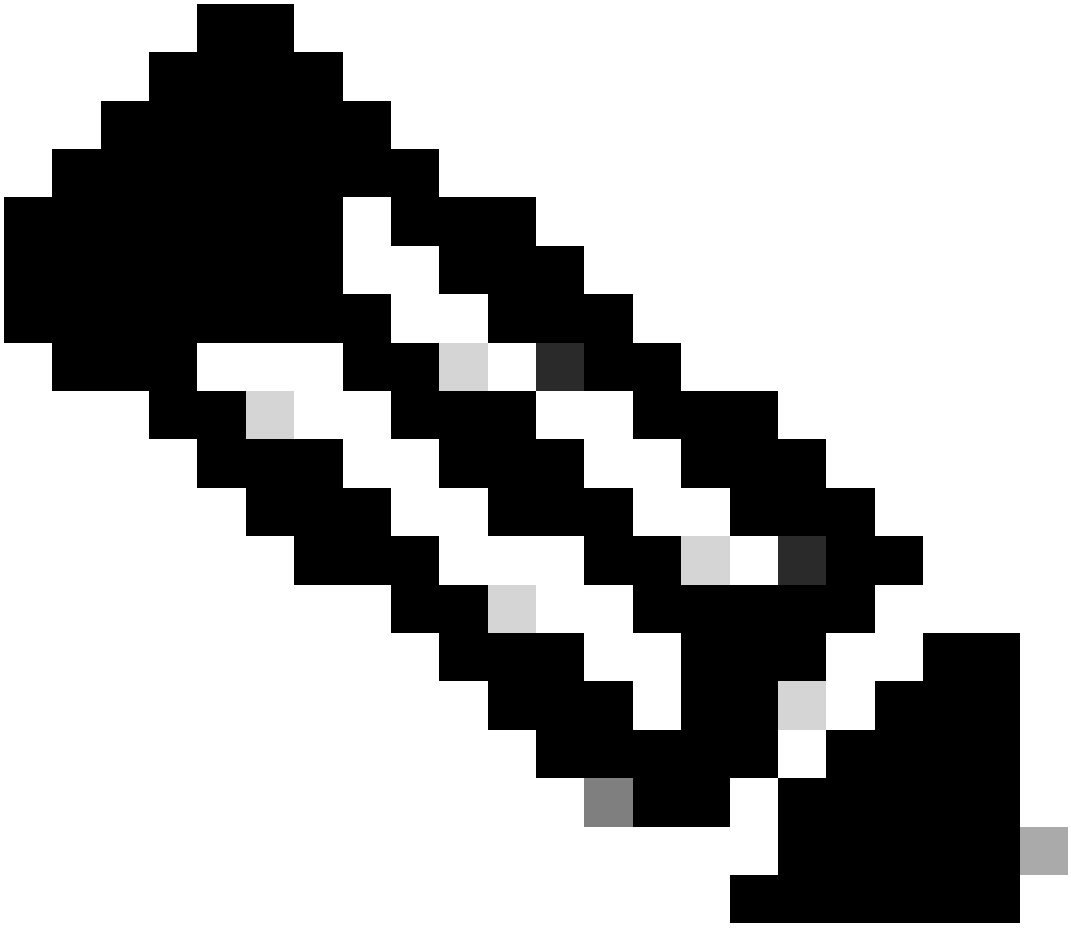
```
SPOKE2#show running-config interface tunnel10
Building configuration...

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
```

Na configuração que você precisa para validar se o mapeamento para o HUB está correto, a string de autenticação NHRP está correspondendo entre os dispositivos, ambos os spokes têm a mesma fase DMVPN configurada e, se a proteção IPSEC for usada, verifique se a configuração de criptografia correta está aplicada.

Se a configuração estiver correta e incluir proteção IPSEC, será necessário verificar se os protocolos IKE e IPSEC estão funcionando corretamente. Isso ocorre porque o NHRP usa o túnel IPSEC como método de transporte para negociar totalmente. Para verificar o estado dos protocolos IKE/IPSEC, execute o comando show crypto IPSEC sa peer x.x.x.x (onde x.x.x.x é o endereço IP NBMA do spoke com o qual você está tentando estabelecer o túnel).



Observação: para verificar se o túnel IPSEC está ativo, a seção ESP (Encapsulation Security Payload) de entrada e saída deve ter as informações de túnel (SPI, transform-set etc.). Todos os valores mostrados nesta seção devem corresponder em ambas as extremidades.



Observação: se forem identificados problemas com IKE/IPSEC, a identificação e solução de problemas deverá se concentrar nesses protocolos.

Status do túnel IKE/IPSEC em 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:

Status do túnel IKE/IPSEC em 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:
```

As saídas mostram que em ambos os spokes o túnel IPSEC está ativo, mas Spoke2 mostra pacotes criptografados (encaps) mas nenhum pacote descriptografado (decaps). Enquanto isso, Spoke1 não mostra nenhum pacote fluindo pelo túnel IPSEC. Isso indica que o problema pode estar no protocolo NHRP.

Ferramentas de identificação e solução de problemas

Depois de fazer a validação inicial e confirmar a configuração e os protocolos IKE/IPSEC (se necessário) não estão causando o problema de comunicação, você pode usar as ferramentas apresentadas nesta seção para continuar a solução de problemas.

Comandos úteis

O comando show dmvpn interface tunnel<ID> fornece informações de sessão específicas de DMVPN (endereços IP de NBMA/túnel, estado do túnel, tempo de atividade/inatividade e atributo). Você pode usar a palavra-chave detail para mostrar detalhes da sessão/soquete de criptografia. É importante mencionar que o estado do túnel deve coincidir em ambas as extremidades.

Saída do spoke 1 show dmvpn interface tunnel<ID>:

<#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
```

Saída do spoke 2 show dmvpn interface tunnel<ID> :

<#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
----- --------------- --------------- ----- -------- -----
```

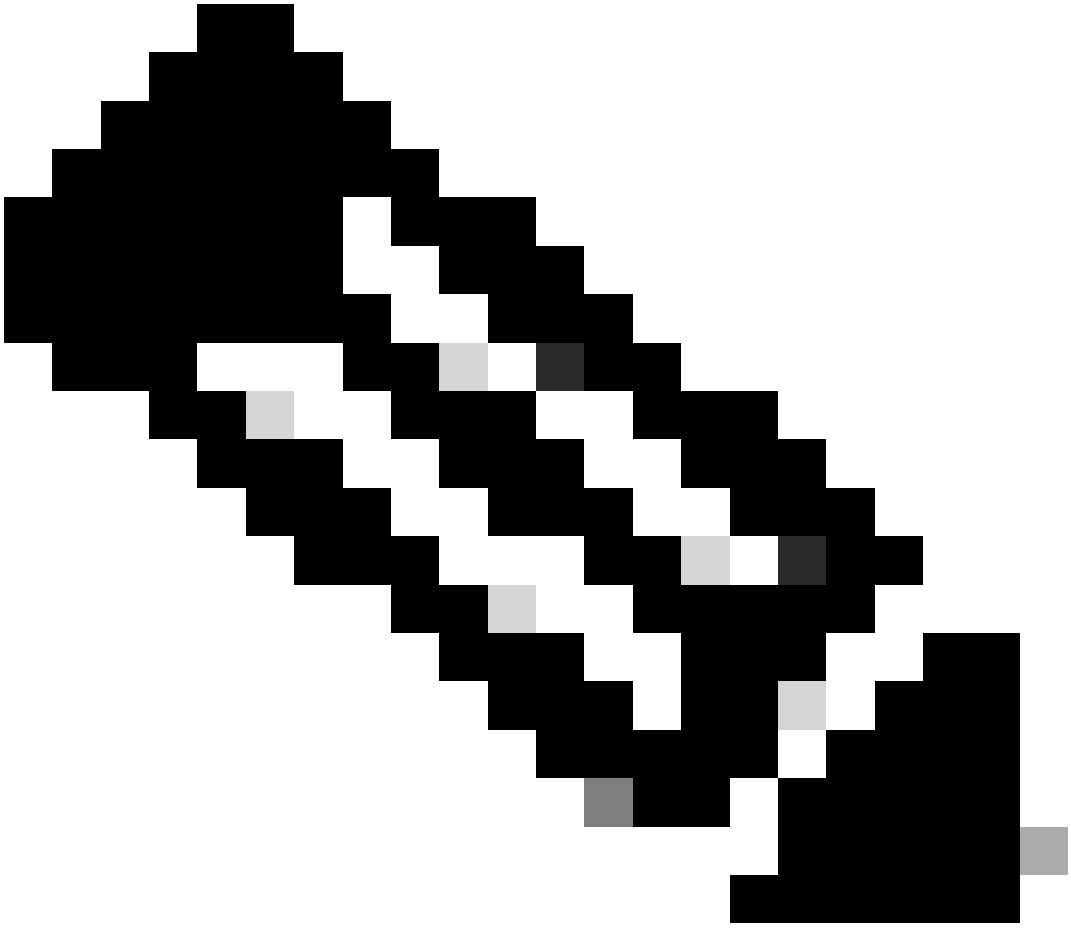
1   172.21.100.1   10.10.10.1   UP 00:03:53   D

```
  1    172.20.10.10     10.10.10.10    UP  02:59:14   S
```

A saída em cada dispositivo mostra informações diferentes para cada spoke. Na tabela Spoke1, você pode ver que a entrada para Spoke 2 não inclui o endereço IP NBMA correto e o atributo parece incompleto (I2). Por outro lado, a tabela Spoke2 mostra o mapeamento correto (endereços IP NBMA/túnel) e o estado como ativo, indicando que o túnel está totalmente negociado.

Os próximos comandos podem ser úteis durante o processo de solução de problemas:

- show ip nhrp: Exibir informações de mapeamento NHRP
- show ip nhrp traffic interface tunnel10: Exibe estatísticas de tráfego NHRP

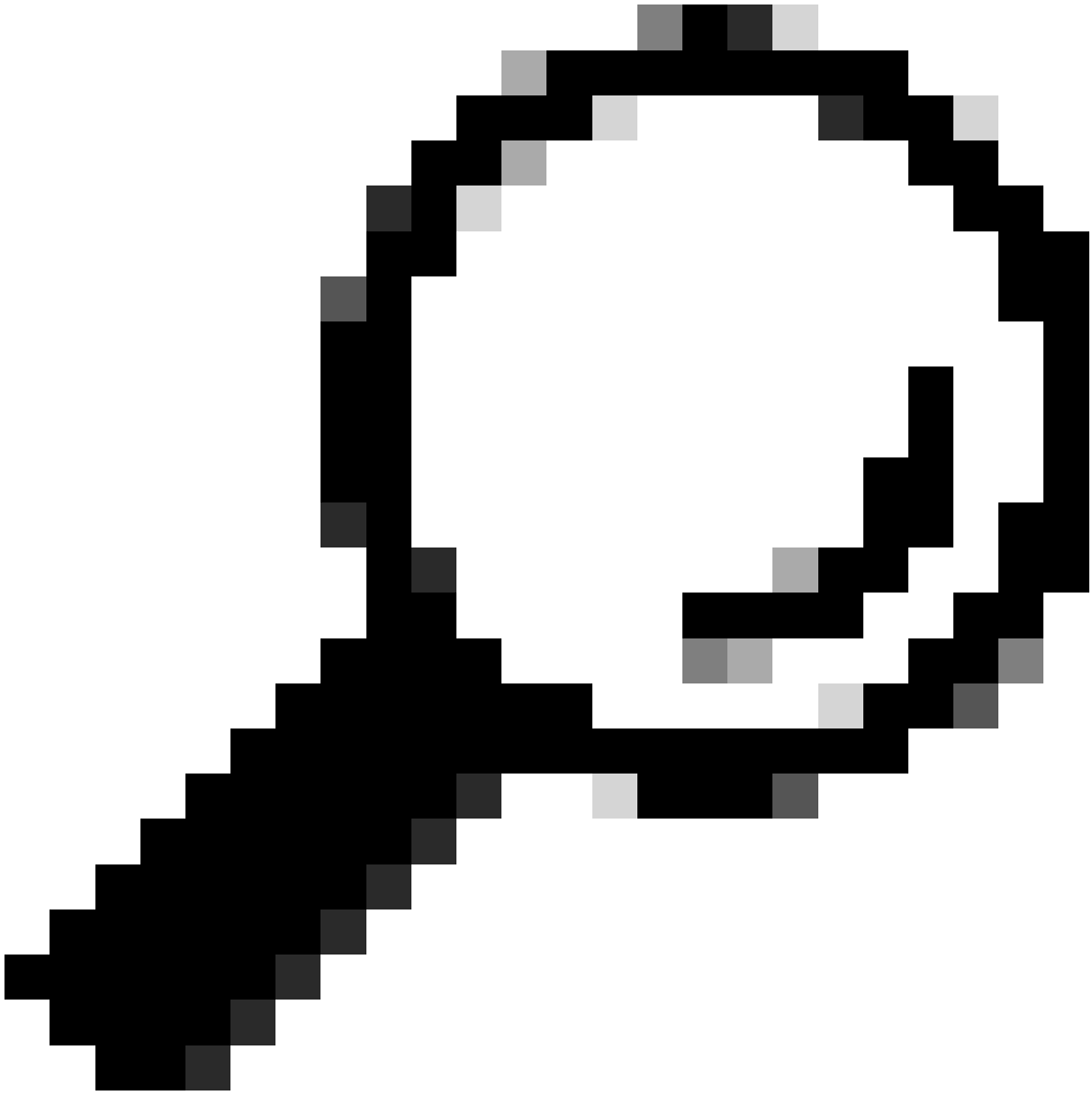


Observação: para especificações de comando (sintaxe, descrição, palavras-chave, exemplo), consulte a Referência de Comando: Referência de Comando de Segurança do Cisco IOS: Comandos S a Z

Debugs

Depois de verificar as informações anteriores e confirmar que o túnel está tendo problemas de negociação, é necessário ativar as depurações para observar como os pacotes NHRP estão sendo trocados. As próximas depurações devem ser ativadas em todos os dispositivos envolvidos:

1. debug dmvpn condition peer NBMA x.x.x.x (onde x.x.x.x é o endereço IP do dispositivo remoto).
2. debug dmvpn all: este comando ativa os comandos de depuração ISAKMP, IKEv2, IPSEC, DMVPN e NHRP.



Dica: é recomendável usar o comando peer condition sempre que você ativar as depurações para que possa ver a negociação desse túnel específico.

Para ver o fluxo NHRP completo, os próximos comandos de depuração foram usados em cada dispositivo:

Spoke1

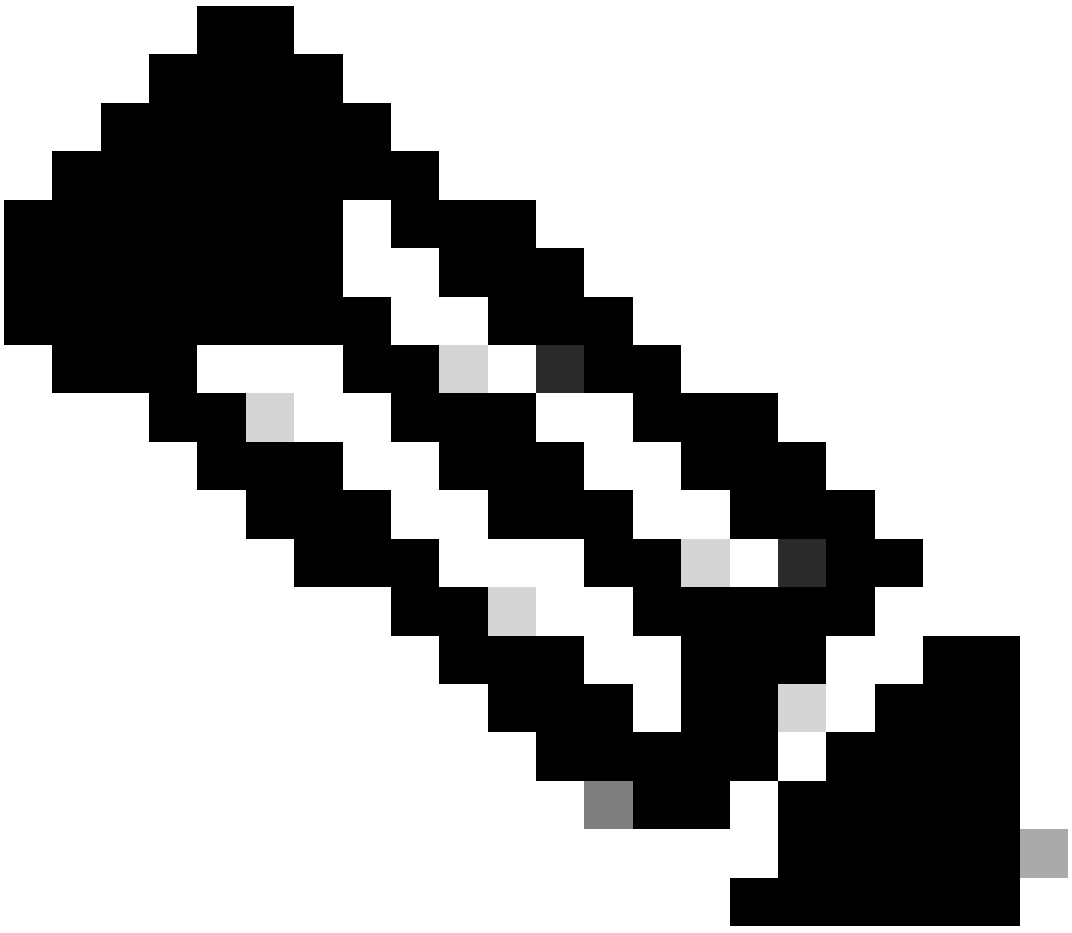
```
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```

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
```



Observação: as depurações devem ser ativadas e coletadas simultaneamente em todos

---

os dispositivos envolvidos.

As depurações habilitadas em todos os dispositivos são exibidas com o 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
```

Depois de coletar todas as depurações, você deve começar a analisar as depurações no spoke de origem (Spoke1), isso permite que você rastreie a negociação desde o início.

Saída de depuração do 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
```

Quando o processo Spoke1 NHRP é iniciado, os registros mostram que o dispositivo está enviando a solicitação de resolução NHRP. O pacote tem algumas informações importantes, como o src NMBA e o protocolo src, que são o endereço IP NBMA e o endereço IP do túnel do spoke de origem (Spoke1). Você também pode ver o valor do protocolo dst que tem o endereço IP do túnel do spoke de destino (Spoke2). Isso indica que Spoke1 está solicitando o endereço NBMA de Spoke2 para concluir o mapeamento. Também no pacote, você pode encontrar o valor reqid que pode ajudá-lo a rastrear o pacote ao longo do caminho. Esse valor permanecerá o mesmo durante todo o processo e pode ser útil para rastrear um fluxo específico da negociação de NHRP. O pacote tem outros valores que são importantes para a negociação, como a string de autenticação NHRP.

Depois que o dispositivo envia a solicitação de resolução NHRP, os registros mostram que uma retransmissão é enviada. Isso ocorre porque o dispositivo não está vendo a resposta de resolução NHRP e, portanto, envia o pacote novamente. Como Spoke1 não está vendo a resposta, é necessário rastrear esse pacote no próximo dispositivo no caminho, significando o HUB.

Saída de depuração de 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

Usando o valor de requid, você pode observar que o HUB recebe a solicitação de resolução enviada por Spoke1. No pacote, os valores de src NBMA e src protocol são as informações de Spoke1, e o valor de dst protocol é o IP de túnel de Spoke2, como visto nas depurações de Spoke1. Quando o HUB recebe a solicitação de resolução, ele executa uma pesquisa de rota e encaminha o pacote para Spoke2. No pacote encaminhado, o HUB adiciona uma extensão contendo suas próprias informações (endereço IP NBMA e endereço IP do túnel).

As depurações anteriores mostram que o HUB está encaminhando corretamente a solicitação de resolução para o spoke 2. Portanto, o próximo passo é confirmar se Spoke2 está recebendo-o, processando-o corretamente e enviando a Spoke1 a resposta de resolução.

Saída de depuração do 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
```

O reqid corresponde ao valor visto nas saídas anteriores; com isso, confirma-se que o pacote de solicitação de resolução NHRP enviado por Spoke1 alcança Spoke2. Esse pacote dispara uma consulta de rota em Spoke2 e percebe que a solicitação de resolução é para si mesmo, portanto, Spoke2 adiciona as informações de Spoke1 à sua tabela NHRP. Antes de enviar o pacote de resposta de resolução de volta para Spoke1, o dispositivo adiciona suas próprias informações (endereço IP NBMA e endereço IP de túnel) para que Spoke1 possa usar esse pacote para adicionar essas informações ao seu banco de dados.

Com base em todas as depurações vistas, o envio de Resposta de Resolução NHRP de Spoke2

não está chegando a Spoke1. O HUB pode ser descartado do problema quando recebe e encaminha o pacote de solicitação de resolução NHRP como esperado. Portanto, o próximo passo é fazer capturas entre Spoke1 e Spoke2 para obter mais detalhes sobre o problema.

Captura de pacotes incorporada

O recurso incorporado de captura de pacotes permite analisar o tráfego que passa pelo dispositivo. A primeira etapa para configurá-lo é criar uma lista de acesso que inclua o tráfego que você deseja capturar em ambos os fluxos de tráfego (entrada e saída).

Para esse cenário, os endereços IP NBMA são usados:

```
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
```

Em seguida, configure a captura usando o comando monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 interface <WAN_INTERFACE> e inicie a captura com o comando monitor capture <CAPTURE_NAME> start.

Capturar configuração em Spoke1 e Spoke2:

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

Para exibir a saída da captura, use o comando show monitor capture <CAPTURE_NAME> buffer brief.

Capturar saída 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
```

Capturar saída Spoke2:

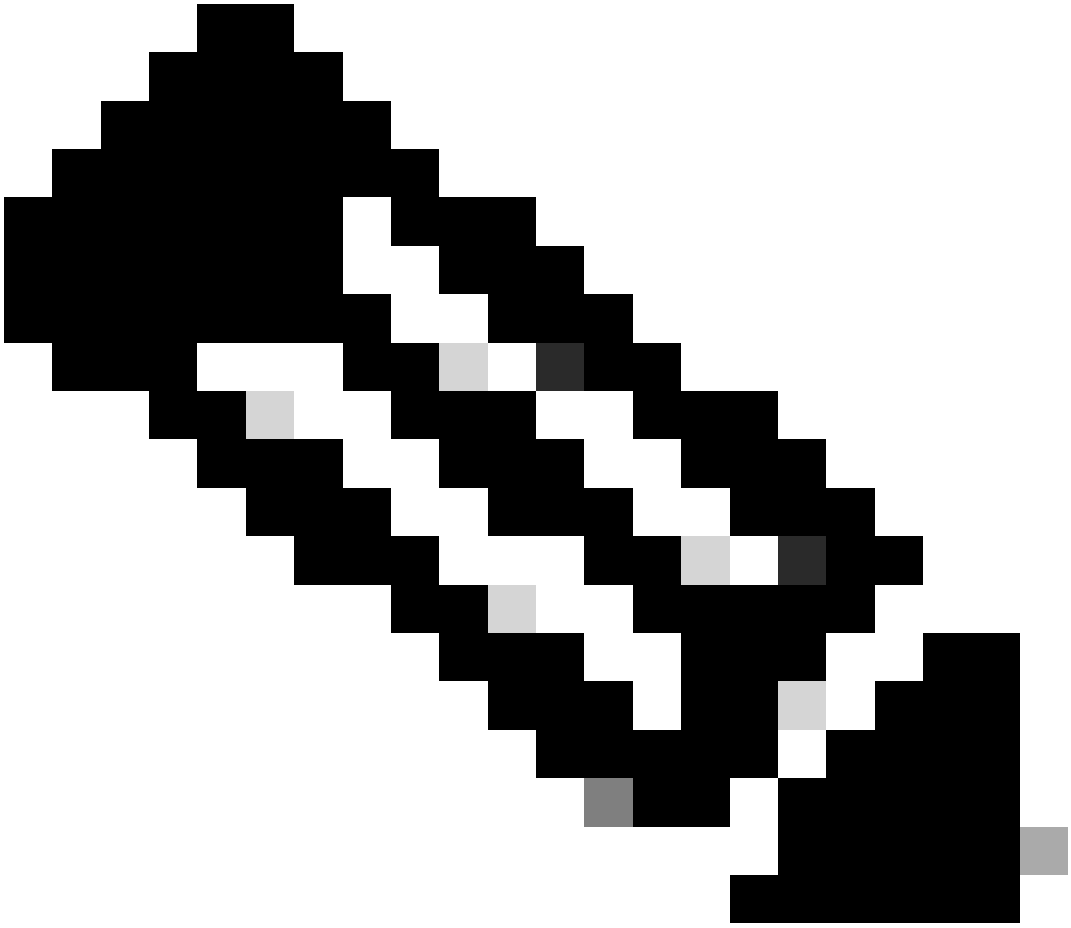
<#root>

```
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
```

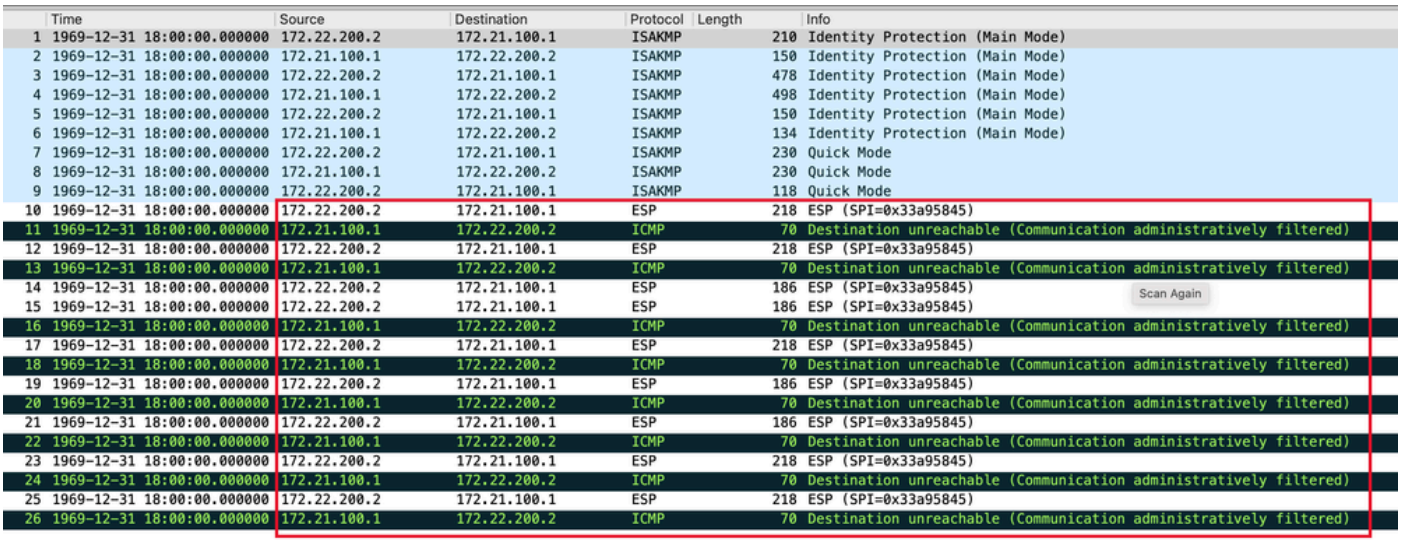
A saída das capturas mostra que os pacotes iniciais são tráfego UDP, indicando a negociação IKE/IPSEC. Depois disso, Spoke2 envia a resposta de resolução para Spoke1, que é visto como tráfego ESP (pacote 9). Depois disso, o fluxo de tráfego esperado é ESP, no entanto, o próximo pacote visto é o tráfego ICMP vindo de Spoke1 para Spoke2.

Para analisar mais profundamente o pacote, você pode exportar o arquivo pcap do dispositivo executando o comando show monitor capture <CAPTURE_NAME> buffer dump. Em seguida, use uma ferramenta de decodificação para converter a saída de despejo em um arquivo pcap para que você possa abri-lo com o Wireshark.



Observação: a Cisco tem um analisador de pacotes onde você pode encontrar a configuração de captura, exemplos e um decodificador: Ferramenta Cisco TAC - Packet Capture Config Generator and Analyzer

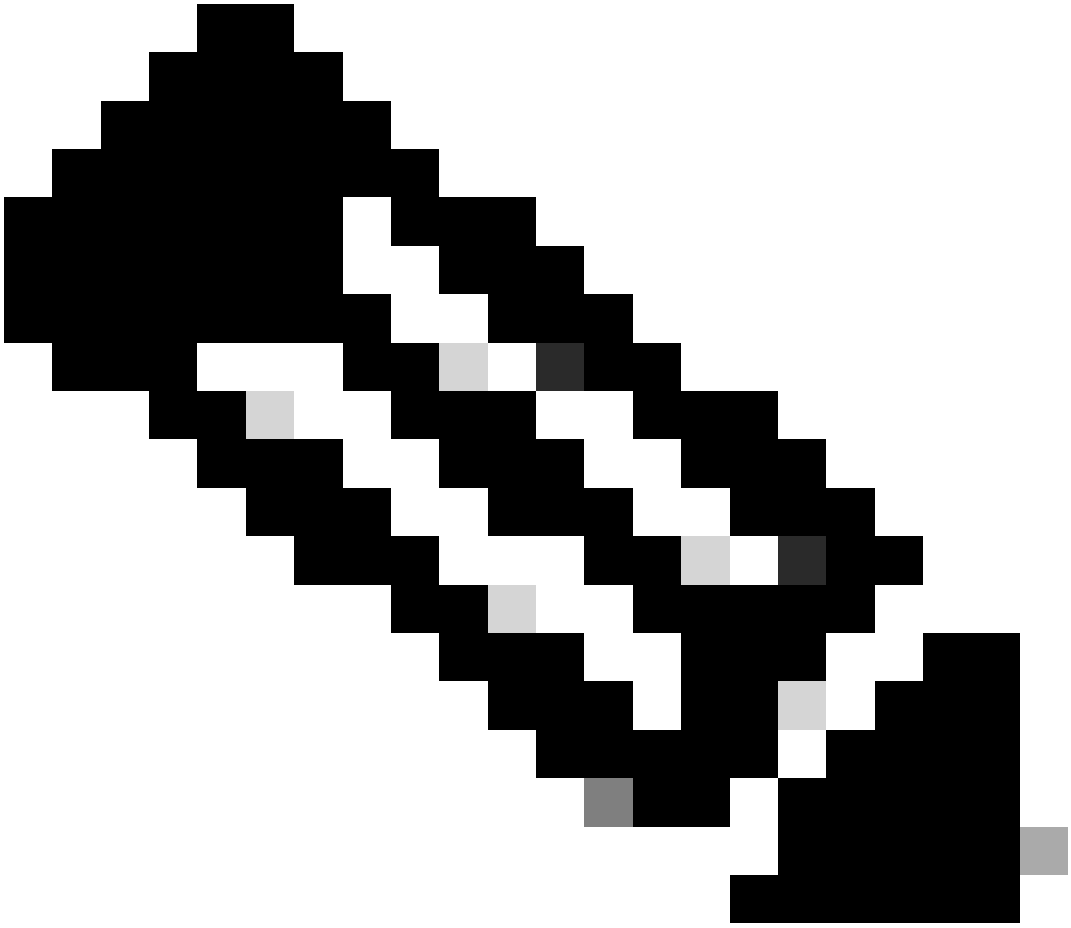
Saída do Wireshark:



	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 saída no Wireshark

O conteúdo do pacote ICMP tem a mensagem de erro Destino inalcançável (comunicação filtrada administrativamente). Isso indica que há algum tipo de filtro, como uma ACL de roteador ou firewall, que afeta o tráfego ao longo do caminho. Na maioria das vezes, o filtro é configurado no dispositivo que envia o pacote (nesse caso, Spoke1), mas os dispositivos intermediários também podem enviá-lo.



Observação: a saída do Wireshark é a mesma em ambos os spokes.

Recurso de rastreamento de pacote de caminho de dados do Cisco IOS® XE

O recurso de rastreamento de pacote de caminho de dados do Cisco IOS XE é usado para analisar como o dispositivo está processando o tráfego. Para configurá-la, você precisa criar uma lista de acesso que inclua o tráfego que deseja capturar em ambos os fluxos de tráfego (entrada e saída).

Para esse cenário, os endereços IP NBMA são usados.

```
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
```

Em seguida, configure o recurso fia-trace e defina a condição de depuração para usar a lista de acesso. Finalmente, inicie a condição.

```
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 o rastreamento de fia detalhado, interrompendo-o quando a quantidade de pacotes configurados tiver sido capturada
- debug platform condition ipv4 access-list <ACL-NAME> both: define uma condição no dispositivo usando a lista de acesso configurada anteriormente
- debug platform condition start: inicia a condição

Para revisar a saída do fia-trace, use os próximos comandos.

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

Saída do spoke1 show platform packet-trace statistics:

<#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
```

Na saída do comando show platform packet-trace statistics, você pode ver os contadores dos pacotes processados pelo dispositivo. Isso permite que você veja os pacotes de entrada e de saída e verifique se o dispositivo está descartando algum pacote, juntamente com o motivo da queda.

Na saída mostrada, Spoke1 está descartando alguns pacotes com a descrição Ipv4Acl. Para analisar ainda mais esses pacotes, o comando show platform packet-trace summary pode ser usado.

Saída de show platform packet-trace summary do Spoke1:

<#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
```

Com essa saída, você pode ver cada pacote que chega e sai do dispositivo, bem como as interfaces de entrada e saída. O status do pacote também é mostrado, indicando se ele foi encaminhado, descartado ou processado internamente (punt).

Neste exemplo, essa saída ajudou a identificar os pacotes que estão sendo descartados pelo dispositivo. Usando o comando show platform packet-trace packet <PACKET_NUMBER>, você pode ver como o dispositivo processa esse pacote específico.

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

<#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>

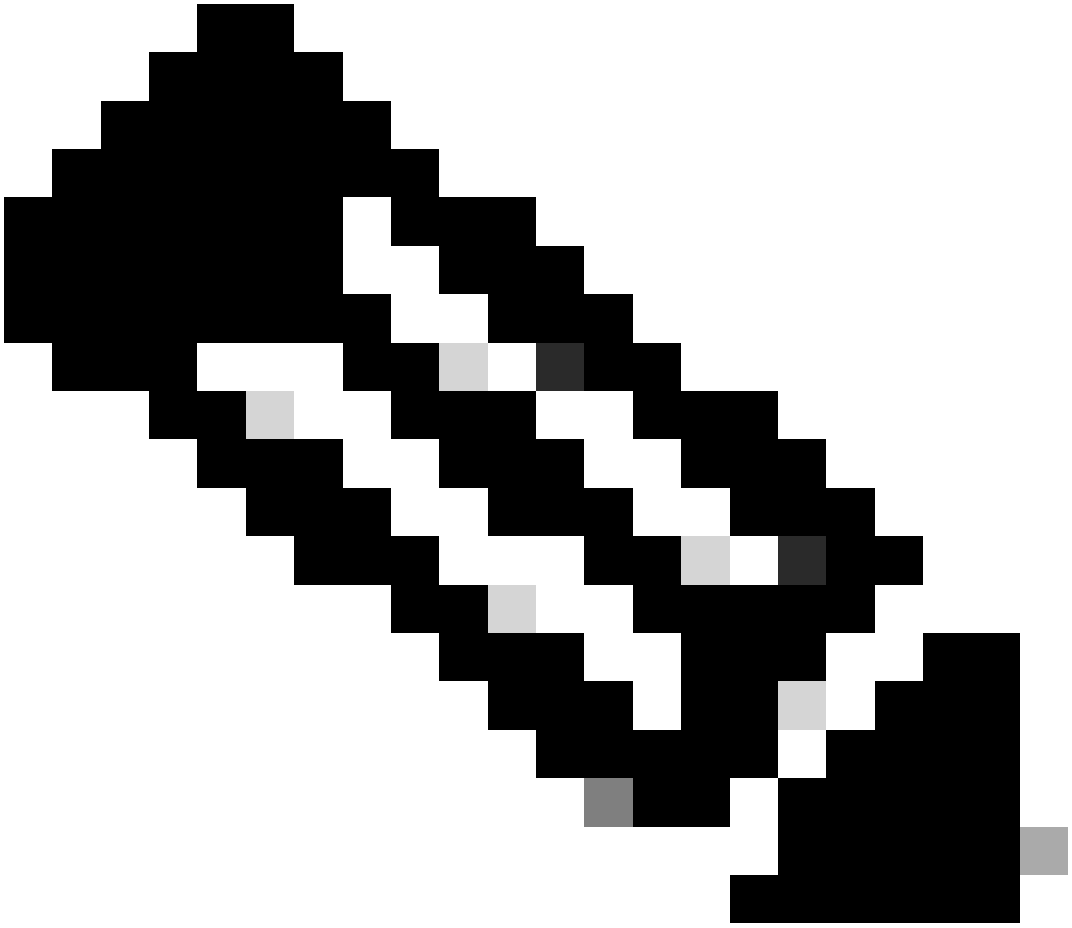
Lapsed time : 174 ns
Feature: INPUT_DROP

```
     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
```

Na primeira parte, você pode ver a interface de entrada e saída e o estado do pacote. Isso é seguido pela segunda parte da saída onde você pode encontrar os endereços IP origem e destino e o protocolo.

Cada fase subsequente mostra como o dispositivo processa esse pacote específico. Isso oferece insights sobre quaisquer configurações como Network Address Translation (NAT) ou access-list ou outros fatores que possam estar afetando-a.

Nesse caso, é possível identificar que o protocolo do pacote é ESP, o IP origem é o endereço IP NBMA de Spoke2 e o IP destino é o endereço IP NBMA de Spoke1. Isso indica que esse é o pacote ausente na negociação de NHRP. Além disso, observa-se que nenhuma interface de saída é especificada em nenhuma fase, sugerindo que algo afetou o tráfego antes que ele pudesse ser encaminhado. Na penúltima fase, você pode ver que o dispositivo está descartando o tráfego de entrada na interface especificada (GigabitEthernet1). A última fase mostra uma lista de acesso de entrada, sugerindo que pode haver alguma configuração na interface que esteja causando a queda.



Observação: se após usar todas as ferramentas de identificação e solução de problemas listadas neste documento, os spokes envolvidos na negociação não estiverem mostrando sinais de que estão descartando ou afetando o tráfego, isso concluirá a solução de problemas nesses dispositivos.

A próxima etapa deve ser verificar os dispositivos intermediários entre eles, como firewalls, switches e ISP.

Solução

Se tal cenário for visto, a próxima etapa é verificar a interface mostrada nas saídas anteriores. Isso envolve a verificação da configuração para verificar se há algo afetando o tráfego.

Configuração da interface 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 sua configuração, a interface tem um grupo de acesso aplicado. É importante verificar se os hosts configurados na lista de acesso não estão interferindo no tráfego usado para a negociação de 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)
```

A segunda instrução da lista de acesso é negar a comunicação entre o endereço IP NBMA de Spoke2 e o endereço IP NBMA de Spoke1, causando a queda vista anteriormente. Após remover o grupo de acesso da interface, a comunicação entre os dois spokes é bem-sucedida:

```
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
```

O túnel IPSEC está ativo e agora está mostrando encapsulamentos e decapitações em ambos os 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:
```

Agora, a tabela DMVPN de Spoke1 está mostrando o mapeamento correto em ambas as 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
```

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.