Dépannage du tunnel satellite de phase 2 DMVPN

Table des matières

Introduction

Ce document décrit comment dépanner un tunnel DMVPN satellite à satellite de phase 2 quand il n'établit pas.

Conditions préalables

Exigences

Cisco recommande que vous ayez des connaissances sur les sujets suivants :

- Réseau privé virtuel multipoint dynamique (DMVPN)
- Protocoles IKE/IPSEC
- Protocole NHRP (Next Hop Resolution Protocol)

Composants utilisés

Ce document est basé sur cette version du logiciel :

- Cisco CSR1000V (VXE) - Version 17.03.08

The information in this document was created from the devices in a specific lab environment. All of

the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Ce document décrit comment configurer et utiliser différents outils de dépannage sur un problème DMVPN courant. Le problème est l'échec de la négociation d'un tunnel DMVPN de phase 2, où le rayon source, l'état DMVPN affiche UP avec le mappage NBMA (Non-Broadcast Multi-Access)/Tunnel correct vers le rayon de destination. Cependant, sur le rayon de destination, un mappage incorrect s'affiche.

Contexte Théorique

Il est important de comprendre comment les tunnels de rayon à rayon sont établis lors de la configuration d'un DMVPN Phase 2. La présente section fournit un bref résumé théorique du processus du PNRH au cours de cette phase.

Dans DMVPN Phase 2, vous pouvez créer des tunnels dynamiques de rayon à rayon à la demande. Cela est possible parce que, sur tous les périphériques du nuage DMVPN (concentrateur et rayons), le mode de l'interface de tunnel passe au multipoint GRE (Generic Routing Encapsulation). L’une des principales caractéristiques de cette phase est que le concentrateur n’est pas perçu comme le saut suivant par les autres périphériques. Au lieu de cela, tous les rayons ont les informations de routage les uns des autres. Lors de l'établissement d'un tunnel de rayon à rayon dans la phase 2, un processus NHRP est déclenché où les rayons apprennent les informations sur d'autres rayons, et effectue un mappage entre les adresses IP de tunnel et NBMA.

Les étapes suivantes indiquent comment le processus de résolution NHRP est déclenché :

1. Lorsque le rayon source tente d'atteindre le réseau local du rayon de destination, il effectue une recherche de route déclenchant le message de demande de résolution pour obtenir l'adresse NBMA du rayon de destination. Le rayon source envoie ce message initial au concentrateur.

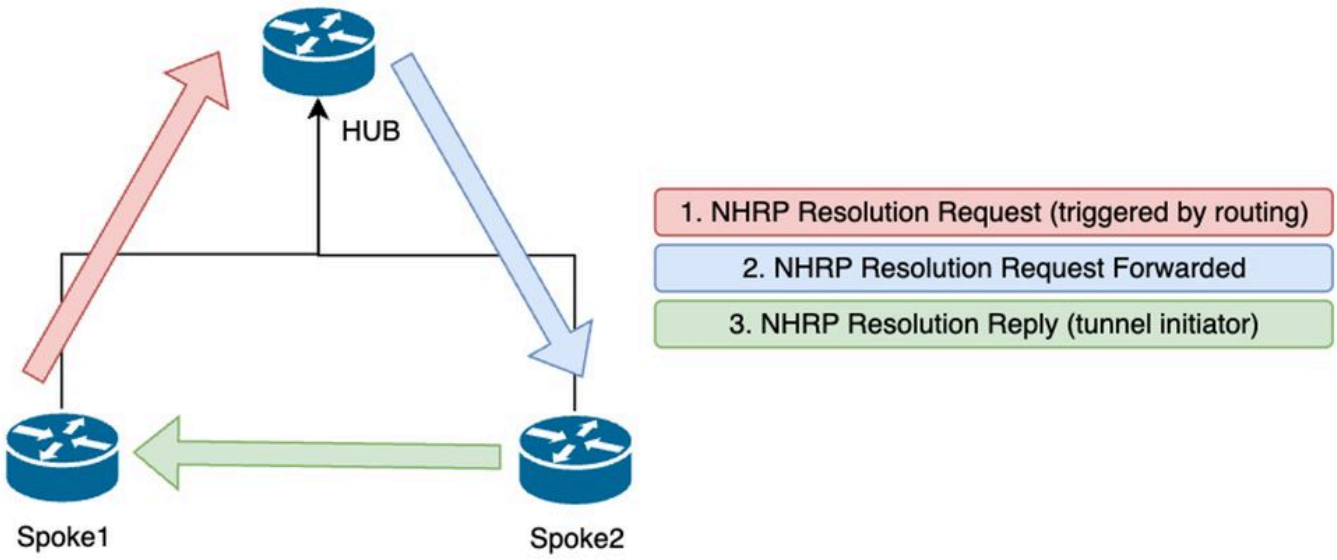
2. Le concentrateur reçoit la demande de résolution et la transmet au rayon de destination.

3. Le rayon de destination envoie la réponse de résolution au rayon source. Si la configuration du tunnel a un profil IPSEC lié :

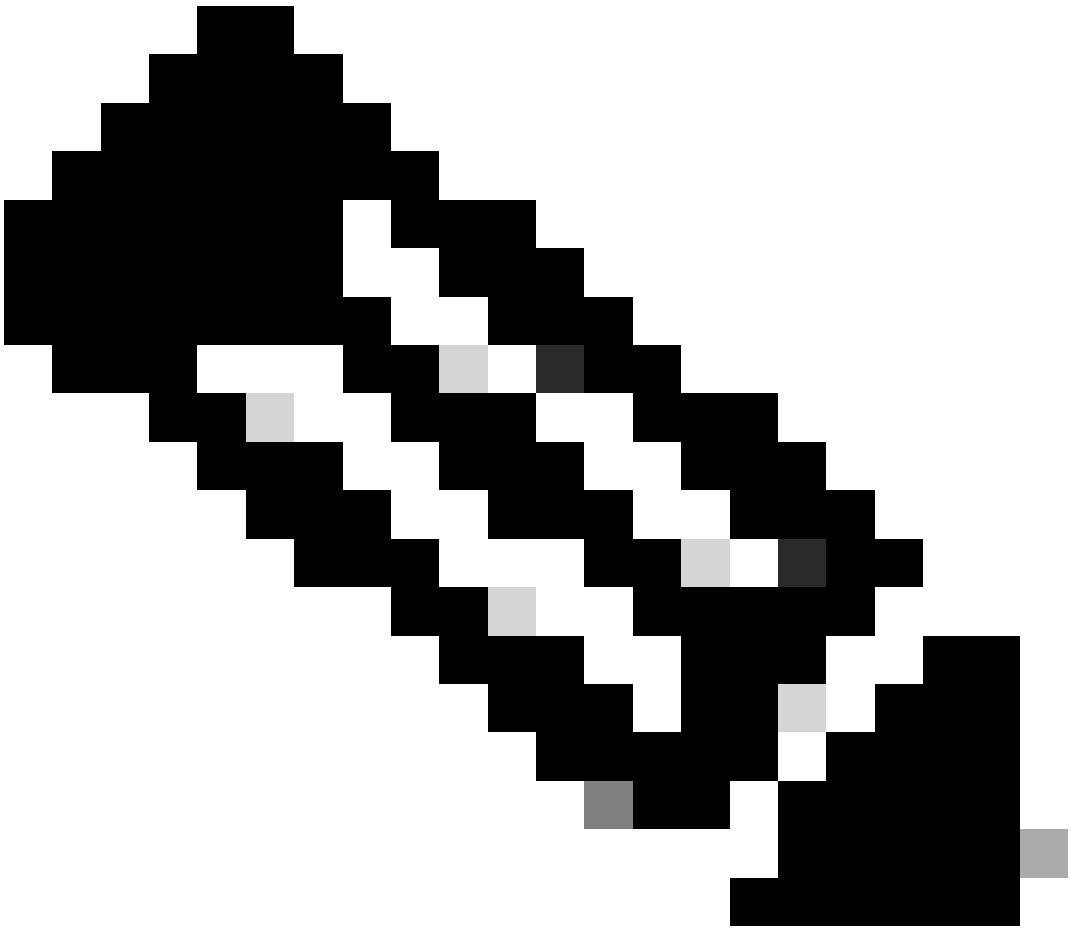
    - Le processus de résolution NHRP est retardé jusqu'à ce que les protocoles IKE/IPSEC puissent s'établir.

    - Le rayon de destination démarre et établit les tunnels IKE/IPSEC.

    - Ensuite, le processus NHRP est repris et le rayon de destination envoie la réponse de résolution au rayon source en utilisant le tunnel IPSEC comme méthode de transport.



Flux de messages NHRP entre les rayons sur la phase 2



Remarque : avant que le processus de résolution puisse démarrer, tous les rayons

---

doivent être déjà enregistrés auprès du concentrateur.

Topologie

Ce schéma montre la topologie utilisée pour le scénario :

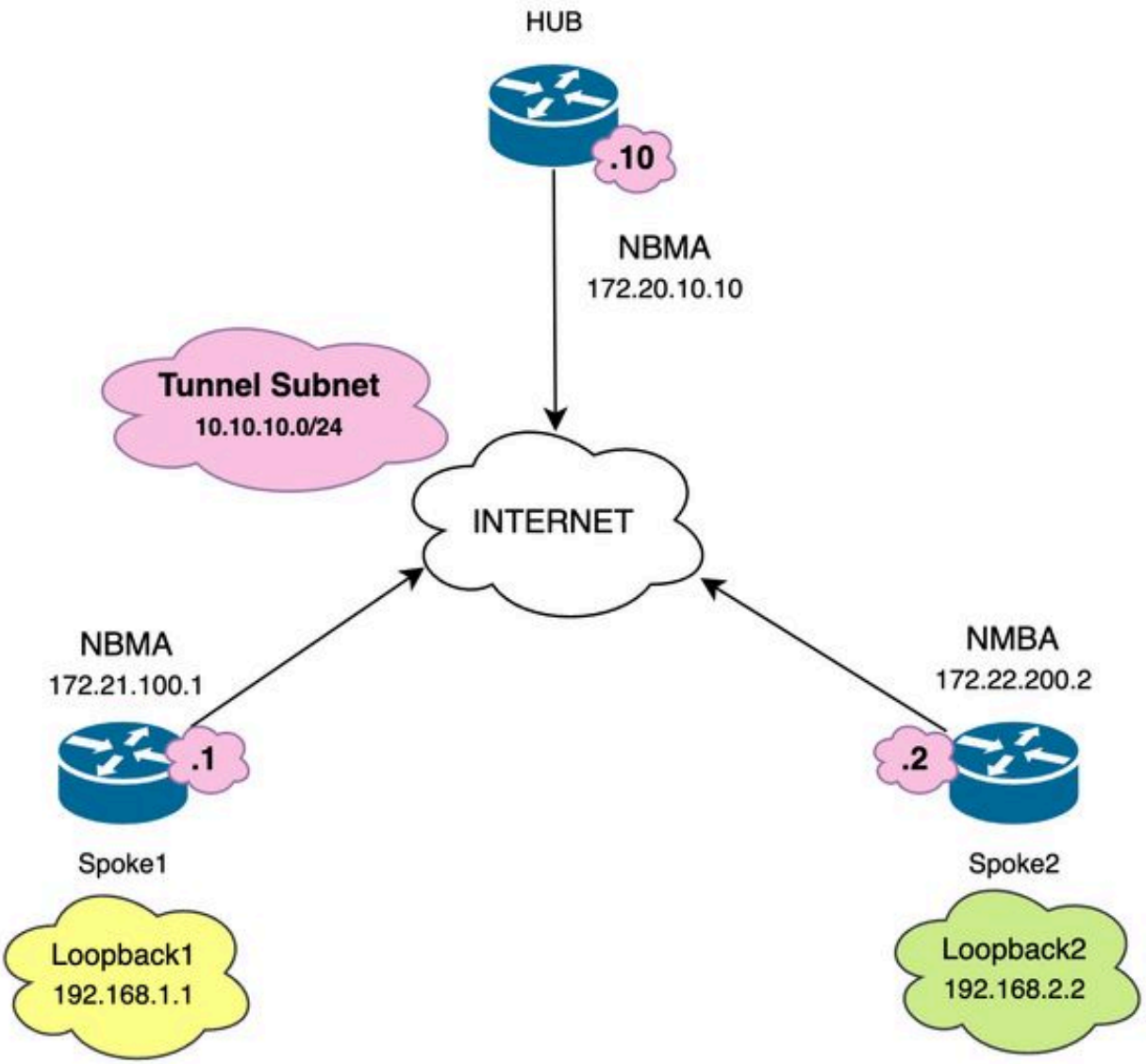


Schéma de réseau et sous-réseaux IP utilisés

Étapes de dépannage

Dans ce scénario, le tunnel de rayon à rayon entre Spoke1 et Spoke2 n'est pas établi, ce qui affecte la communication entre leurs ressources locales (représentées par des interfaces de bouclage) car elles ne peuvent pas se joindre.

```
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)
```

Validation initiale

Lorsque vous rencontrez un tel scénario, il est important de commencer par valider la configuration du tunnel et de s'assurer que les deux périphériques ont les valeurs correctes à l'intérieur. Pour examiner la configuration du tunnel, exécutez la commande show running-config interface tunnel<ID>.

Configuration du tunnel 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
```

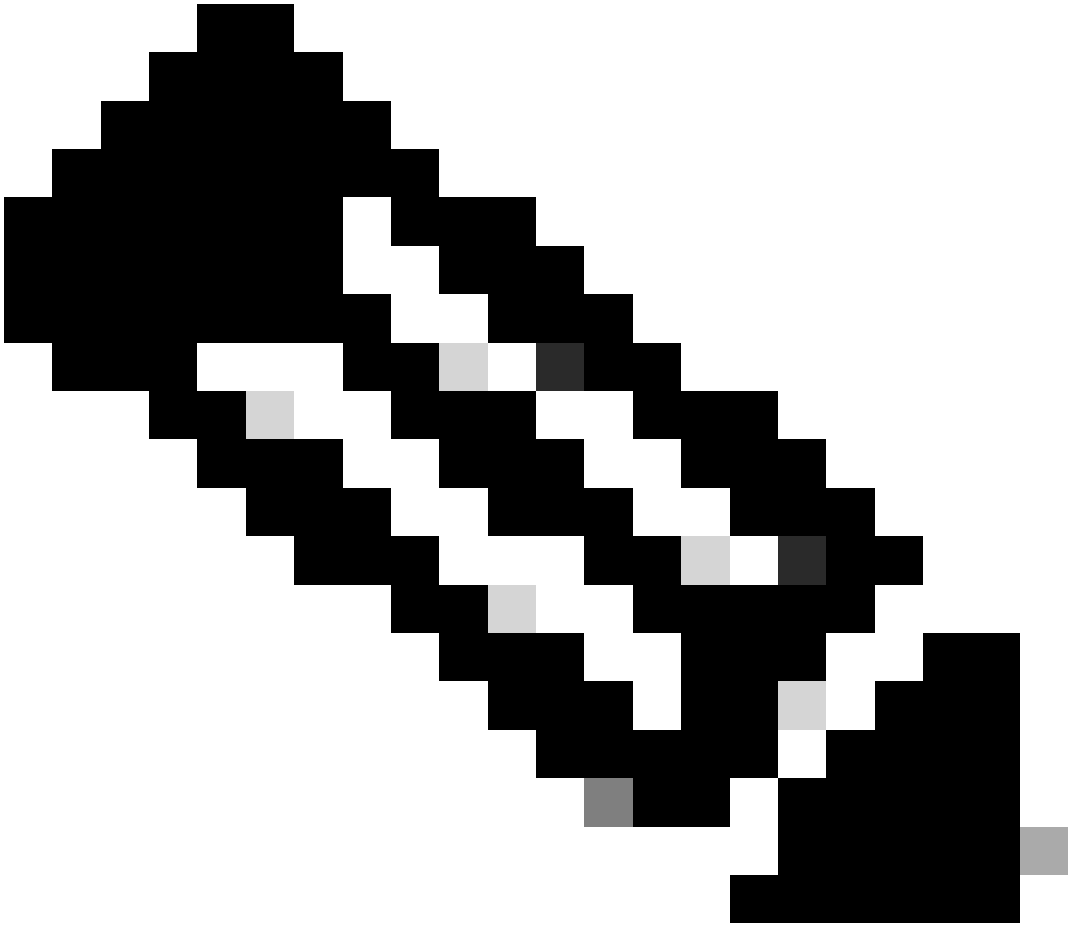
Configuration du tunnel Spoke 2 :

<#root>

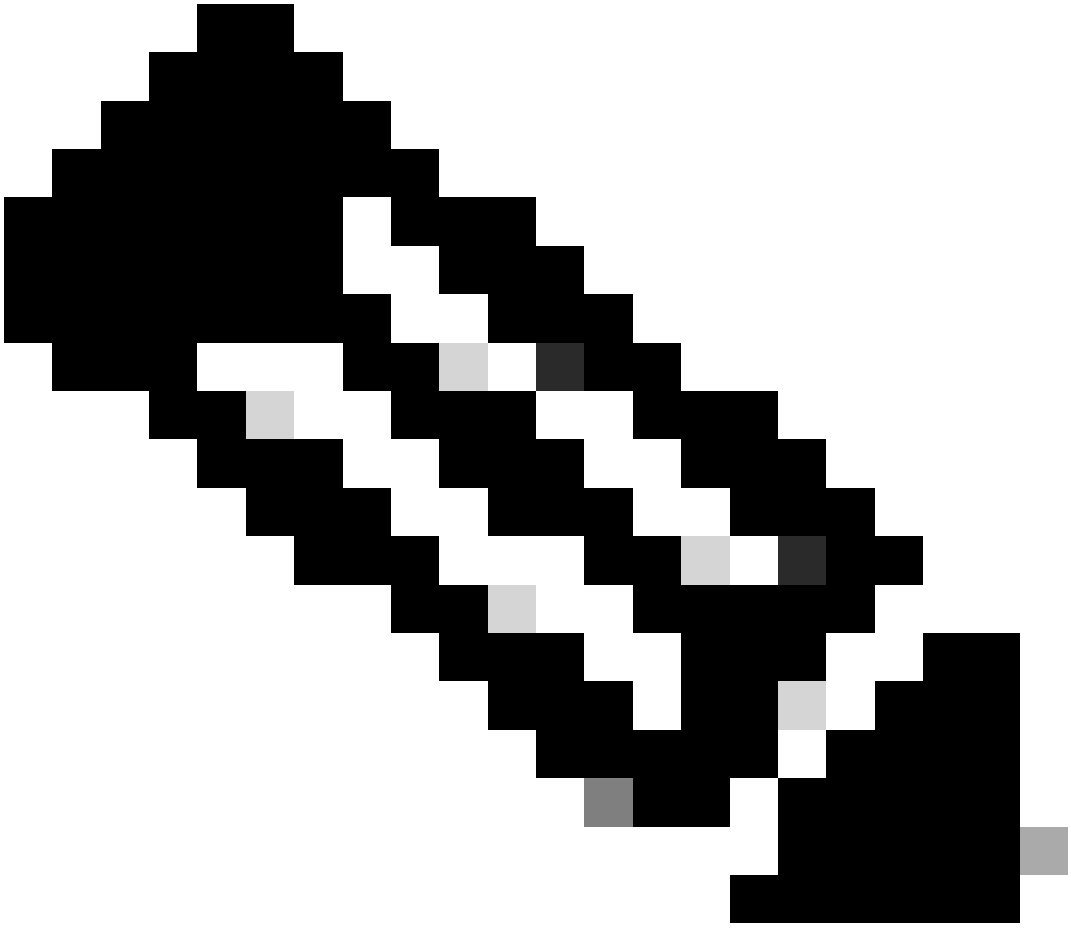
```
SPOKE2#show running-config interface tunnel10
Building configuration...

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
```

Sur la configuration, vous devez valider que le mappage vers le concentrateur est correct, que la chaîne d'authentification NHRP correspond entre les périphériques, que les deux rayons ont la même phase DMVPN configurée et, si la protection IPSEC est utilisée, vérifiez que la configuration de chiffrement correcte est appliquée.

Si la configuration est correcte et qu'elle inclut la protection IPSEC, il est nécessaire de vérifier que les protocoles IKE et IPSEC fonctionnent correctement. En effet, NHRP utilise le tunnel IPSEC comme méthode de transport pour négocier entièrement. Pour vérifier l'état des protocoles IKE/IPSEC, exécutez la commande show crypto IPSEC sa peer x.x.x.x (où x.x.x.x est l'adresse IP NBMA du rayon avec lequel vous essayez d'établir le tunnel).



Remarque : pour vérifier si le tunnel IPSEC est actif, la section Encapsulation Security Payload (ESP) entrante et sortante doit avoir les informations de tunnel (SPI, transform-set, etc.). Toutes les valeurs affichées dans cette section doivent correspondre aux deux extrémités.



Remarque : si des problèmes liés à IKE/IPSEC sont identifiés, le dépannage doit se concentrer sur ces protocoles.

État du tunnel IKE/IPSEC sur 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:

État du tunnel IKE/IPSEC sur 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:
```

Les résultats indiquent que le tunnel IPSEC est actif sur les deux rayons, mais que Spoke2 affiche des paquets chiffrés (encaps) mais aucun paquet déchiffré (decaps). En attendant, Spoke1 n'affiche aucun paquet circulant dans le tunnel IPSEC. Cela indique que le problème peut provenir du protocole NHRP.

Outils de dépannage

Après avoir effectué la validation initiale et confirmé que la configuration et les protocoles IKE/IPSEC (le cas échéant) ne sont pas à l'origine du problème de communication, vous pouvez utiliser les outils présentés dans cette section pour poursuivre le dépannage.

Commandes utiles

La commande show dmvpn interface tunnel<ID> vous donne des informations de session spécifiques à DMVPN (adresses IP NBMA/Tunnel, état du tunnel, temps de fonctionnement/arrêt et attribut). Vous pouvez utiliser le mot clé detail pour afficher les détails de la session/socket de chiffrement. Il est important de mentionner que l'état du tunnel doit correspondre aux deux extrémités.

Résultat de la commande satellite 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
```

Sortie de la commande satellite 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
----- --------------- --------------- ----- -------- -----
```

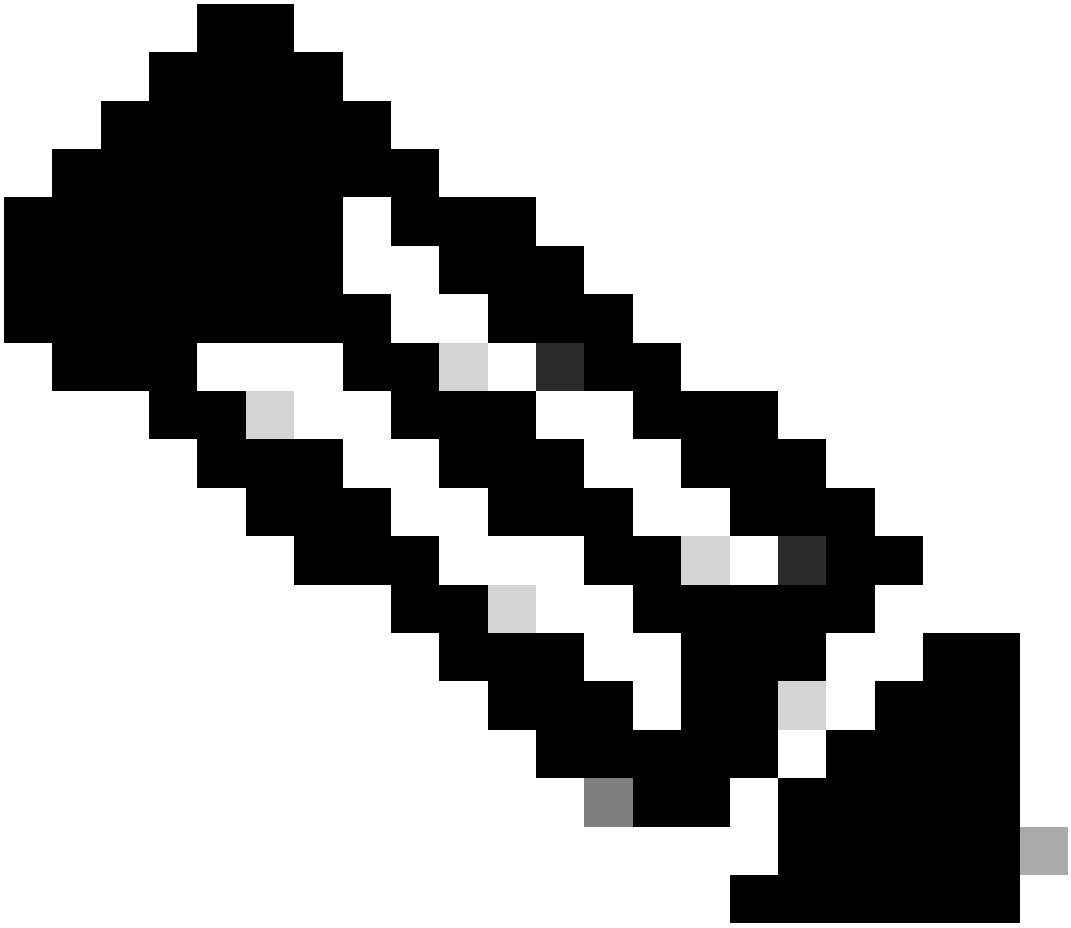
1   172.21.100.1   10.10.10.1   UP 00:03:53   D

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

La sortie de chaque périphérique affiche des informations différentes pour chaque rayon. Dans le tableau Spoke1, vous pouvez voir que l'entrée pour Spoke 2 n'inclut pas l'adresse IP NBMA correcte et que l'attribut semble incomplet (I2). D'autre part, la table Spoke2 affiche le mappage correct (adresses IP NBMA/Tunnel) et l'état up indiquant que le tunnel est entièrement négocié.

Les commandes suivantes peuvent être utiles lors du processus de dépannage :

- show ip nhrp : afficher les informations de mappage NHRP
- show ip nhrp traffic interface tunnel10 : affiche les statistiques de trafic NHRP

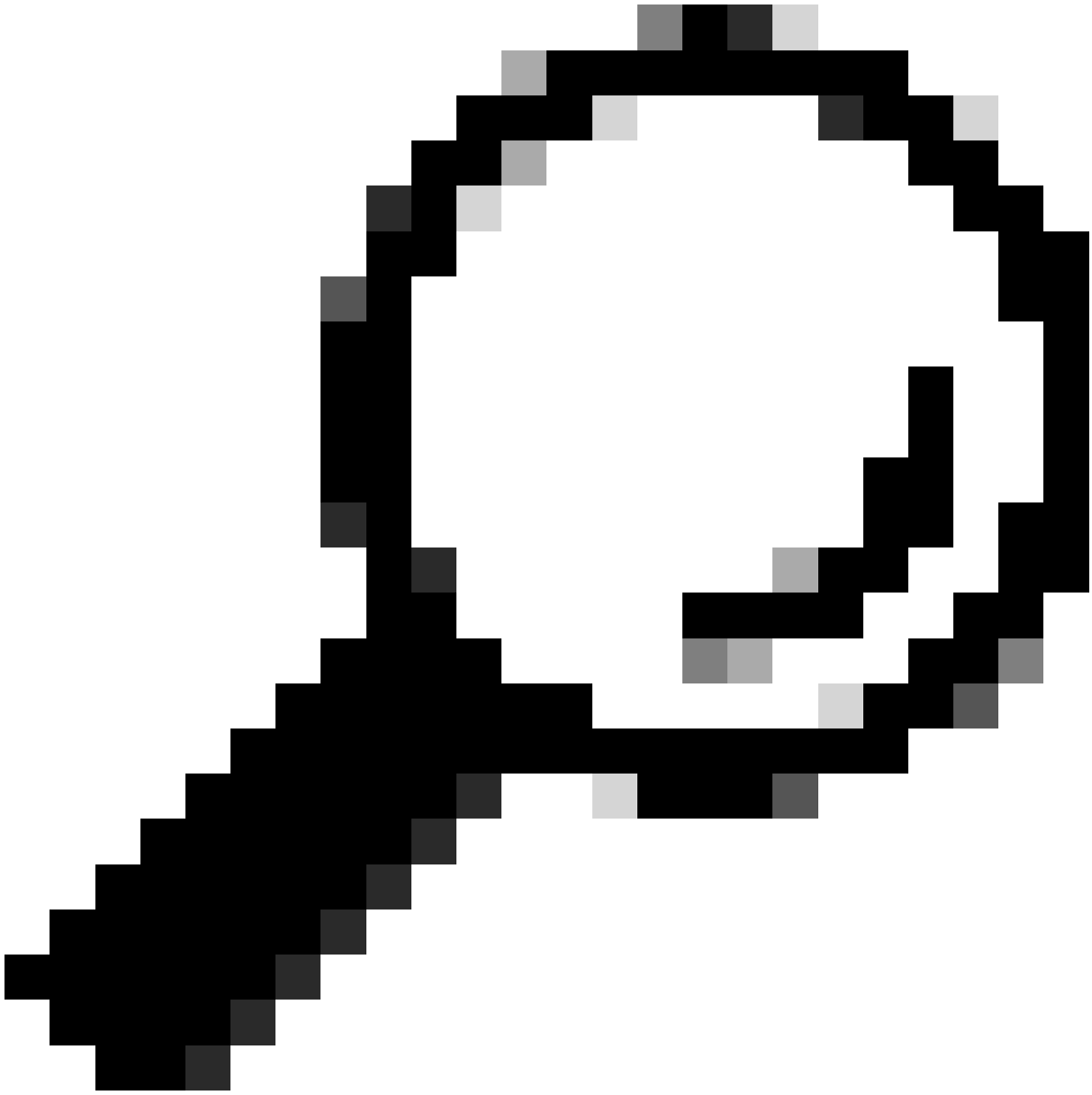


Remarque : pour connaître les spécifications des commandes (syntaxe, description, mots-clés, exemple), consultez le document Command Reference : Cisco IOS Security Command Reference : Commands S to Z

Déboguages

Après avoir vérifié les informations précédentes et confirmé que le tunnel rencontre des problèmes de négociation, il est nécessaire d'activer les débogages pour observer comment les paquets NHRP sont échangés. Les débogages suivants doivent être activés sur tous les périphériques impliqués :

1. debug dmvpn condition peer NBMA x.x.x.x (où x.x.x.x est l'adresse IP du périphérique distant).
2. debug dmvpn all : cette commande active les commandes de débogage ISAKMP, IKEv2, IPSEC, DMVPN et NHRP.



Conseil : il est recommandé d'utiliser la commande peer condition chaque fois que vous activez les débogages afin de voir la négociation de ce tunnel spécifique.

Pour afficher le flux NHRP complet, les commandes debugs suivantes ont été utilisées sur chaque périphérique :

Rayon1

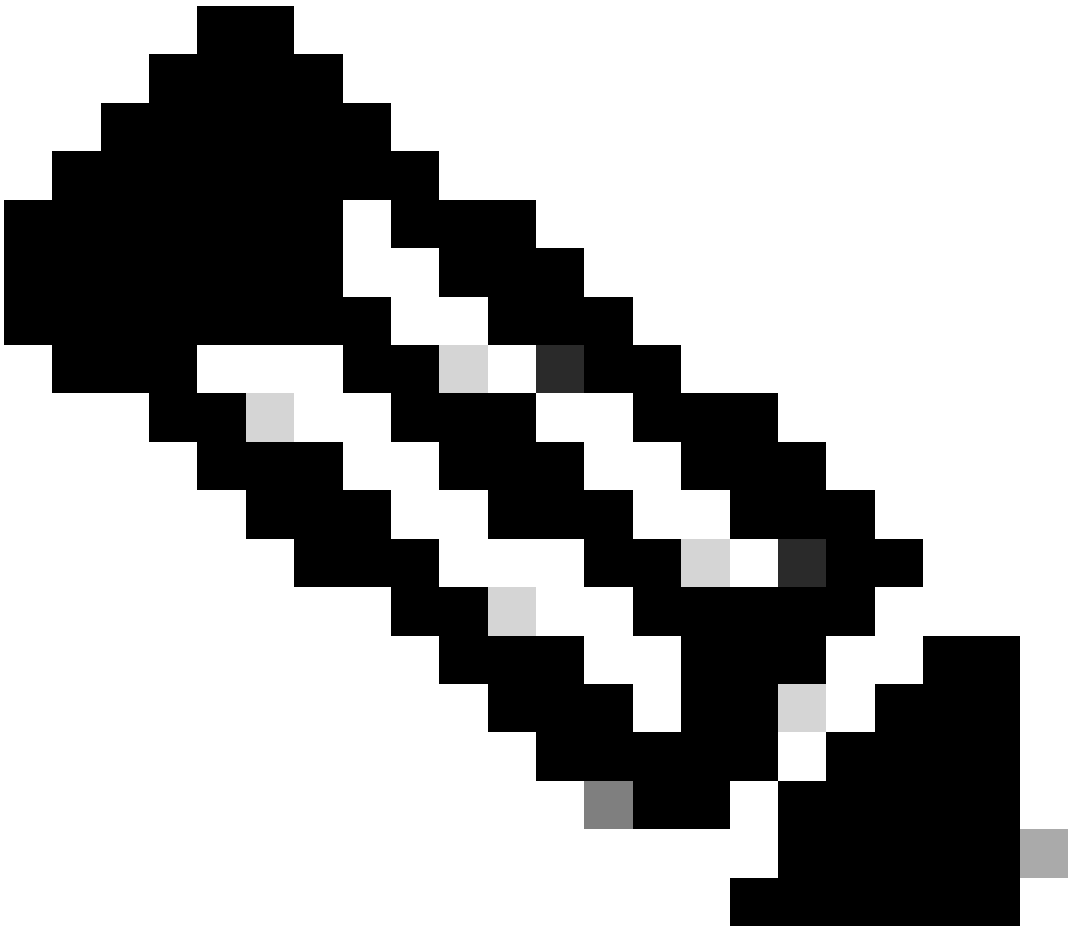
```
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```

CONCENTRATEUR

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

Rayon 2

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



Remarque : les débogages doivent être activés et collectés simultanément sur tous les

---

périphériques impliqués.

Les débogages activés sur tous les périphériques sont affichés avec la commande 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
```

Après avoir collecté tous les débogages, vous devez commencer à analyser les débogages sur le rayon source (Spoke1), ce qui vous permet de tracer la négociation depuis le début.

Sortie de débogage 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
```

Une fois le processus Spoke1 NHRP démarré, les journaux indiquent que le périphérique envoie la demande de résolution NHRP. Le paquet a des informations importantes comme le src NMBA et le protocole src qui sont l'adresse IP NBMA et l'adresse IP de tunnel du rayon source (Spoke1). Vous pouvez également voir la valeur du protocole dst qui a l'adresse IP du tunnel du rayon de destination (Spoke2). Cela indique que Spoke1 demande l'adresse NBMA de Spoke2 pour terminer le mappage. Vous pouvez également trouver sur le paquet la valeur requise qui peut vous aider à suivre le paquet le long du chemin. Cette valeur restera la même tout au long du processus et peut être utile pour suivre un flux spécifique de la négociation NHRP. Le paquet a d'autres valeurs qui sont importantes pour la négociation comme la chaîne d'authentification NHRP.

Une fois que le périphérique a envoyé la demande de résolution NHRP, les journaux indiquent qu'une retransmission est envoyée. Cela est dû au fait que le périphérique ne voit pas la réponse de résolution NHRP et qu'il envoie à nouveau le paquet. Comme Spoke1 ne voit pas la réponse, il est nécessaire de suivre ce paquet sur le périphérique suivant sur le chemin, c'est-à-dire le concentrateur.

Sortie de débogage du concentrateur :

<#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

En utilisant la valeur de reqid, vous pouvez observer que le HUB reçoit la requête de résolution envoyée par Spoke1. Dans le paquet, les valeurs de src NBMA et de src protocol sont les informations de Spoke1, et la valeur de dst protocol est le tunnel IP de Spoke2, comme il a été vu sur les débogages de Spoke1. Lorsque le concentrateur reçoit la demande de résolution, il effectue une recherche de route et transfère le paquet à Spoke2. Dans le paquet transféré, le concentrateur ajoute une extension contenant ses propres informations (adresse IP NBMA et adresse IP du tunnel).

Les débogages précédents montrent que le HUB transfère correctement la demande de résolution vers le satellite 2. Par conséquent, l'étape suivante consiste à confirmer que Spoke2 le reçoit, le traite correctement et envoie à Spoke1 la réponse de résolution.

Sortie de débogage 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: glob
*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
*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 IPSE
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F1
*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 80007F1
*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 d
*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 IPSE
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F1
*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_tim
*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.
*Feb 1 01:31:34.653: NHRP-CACHE: Converted internal dynamic cache entry for 10.10.10.1/32 interface Tun
*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
```

Le requid correspond à la valeur vue dans les sorties précédentes, avec ceci, il est confirmé que le paquet de requête de résolution NHRP envoyé par Spoke1 atteint Spoke2. Ce paquet déclenche une recherche de route sur Spoke2 et réalise que la demande de résolution est pour lui-même. Par conséquent, Spoke2 ajoute les informations de Spoke1 à sa table NHRP. Avant de renvoyer le paquet de réponse de résolution à Spoke1, le périphérique ajoute ses propres informations (adresse IP NBMA et adresse IP de tunnel) afin que Spoke1 puisse utiliser ce paquet pour ajouter ces informations à sa base de données.

D'après tous les débogages observés, la réponse de résolution NHRP envoyée par Spoke2

n'arrive pas à Spoke1. Le concentrateur peut être supprimé du problème lors de la réception et du transfert du paquet de demande de résolution NHRP comme prévu. Par conséquent, l'étape suivante consiste à effectuer des captures entre Spoke1 et Spoke2 pour obtenir plus de détails sur le problème.

Capture de paquets intégrée

La fonction de capture de paquets intégrée vous permet d'analyser le trafic passant par le périphérique. La première étape de la configuration consiste à créer une liste de contrôle d'accès incluant le trafic que vous souhaitez capturer sur les deux flux de trafic (entrant et sortant).

Pour ce scénario, les adresses IP NBMA sont utilisées :

```
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
```

Configurez ensuite la capture à l'aide de la commande monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 interface <WAN_INTERFACE> et démarrez la capture à l'aide de la commande monitor capture <CAPTURE_NAME> start.

Capturer la configuration sur Spoke1 et Spoke2 :

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

Pour afficher le résultat de la capture, utilisez la commande show monitor capture <CAPTURE_NAME> buffer brief.

Capturer la sortie 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
```

Capturer la sortie Spoke2 :

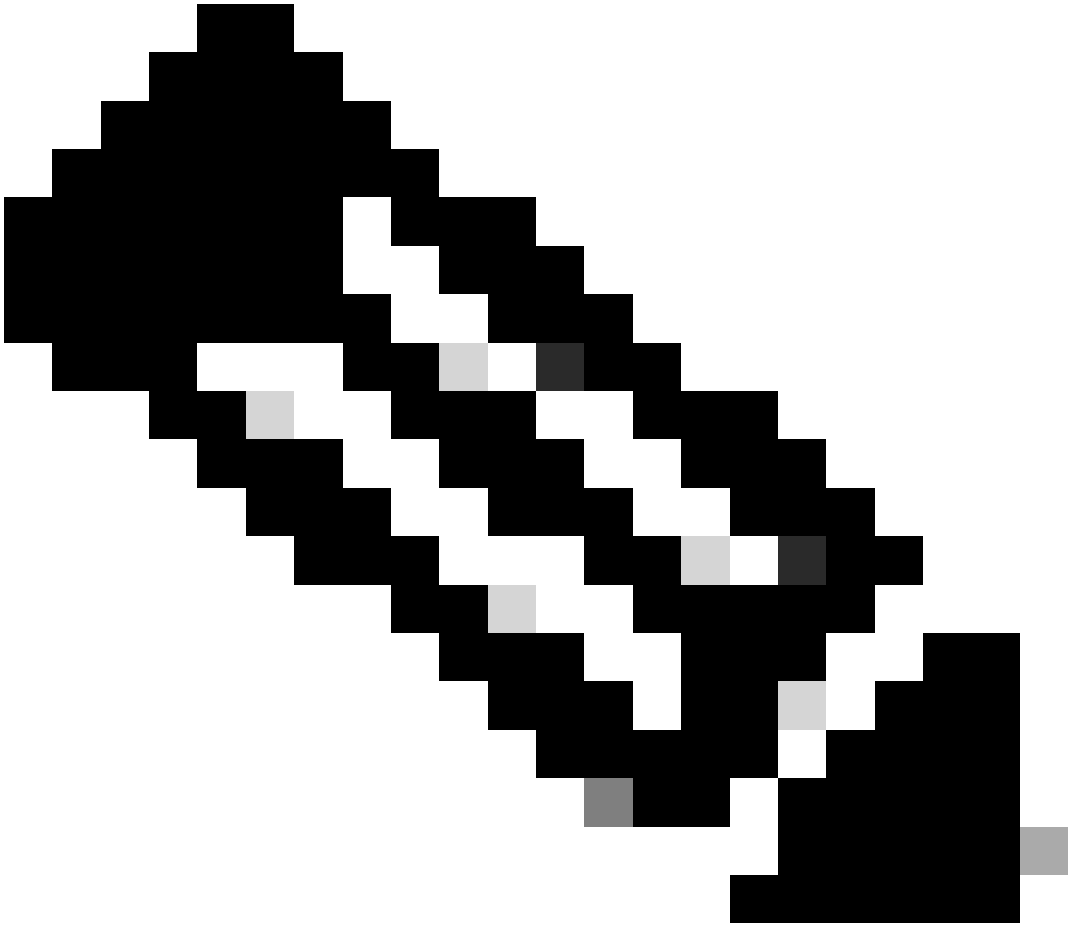
<#root>

```
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
```

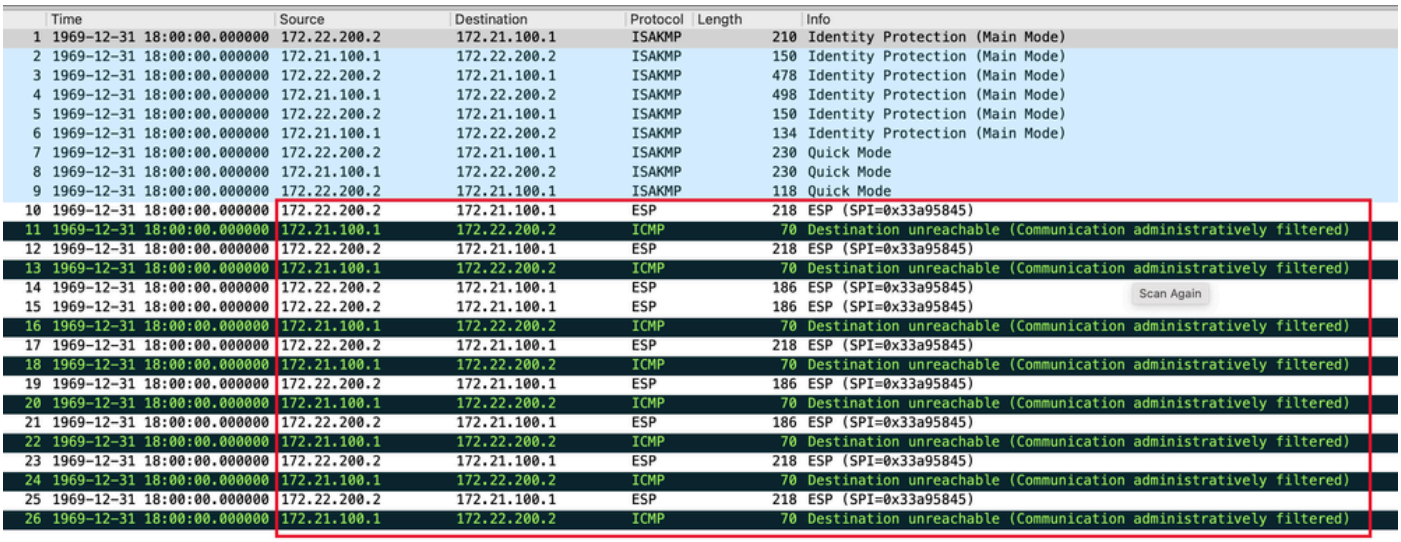
Le résultat des captures indique que les paquets initiaux sont du trafic UDP, indiquant la négociation IKE/IPSEC. Ensuite, Spoke2 envoie la réponse de résolution à Spoke1, qui est considéré comme du trafic ESP (paquet 9). Après cela, le flux de trafic attendu est ESP, cependant, le prochain paquet vu est le trafic ICMP en provenance de Spoke1 vers Spoke2.

Pour analyser plus en profondeur le paquet, vous pouvez exporter le fichier pcap à partir du périphérique en exécutant la commande show monitor capture <CAPTURE_NAME> buffer dump. Ensuite, utilisez un outil de décodage pour convertir la sortie de vidage dans un fichier pcap afin de pouvoir l'ouvrir avec Wireshark.



Remarque : Cisco dispose d'un analyseur de paquets où vous pouvez trouver la configuration de capture, des exemples et un décodeur : Outil TAC Cisco - Générateur et analyseur de configuration de capture de paquets

Sortie Wireshark :

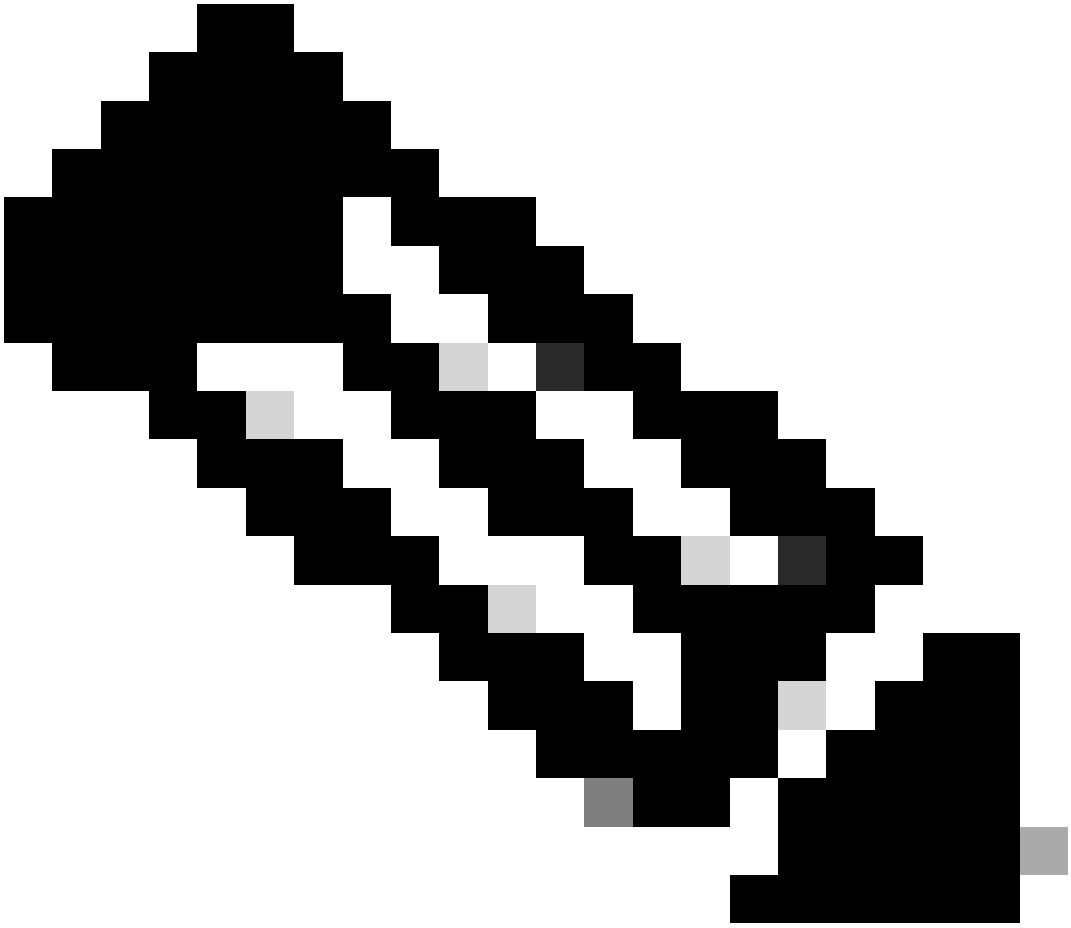


	Time	Source	Destination	Protocol	Length	Info
1	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	210	Identity Protection (Main Mode)
2	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	150	Identity Protection (Main Mode)
3	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	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)

Scan Again

Capturer la sortie sur Wireshark

Le message d’erreur Destination inaccessible (Communication filtrée administrativement) s’affiche dans le contenu du paquet ICMP. Cela indique qu’il existe un filtre, tel qu’une liste de contrôle d’accès de routeur ou un pare-feu, qui affecte le trafic sur le chemin. La plupart du temps, le filtre est configuré sur le périphérique qui envoie le paquet (dans ce cas, Spoke1), mais les périphériques du milieu peuvent également l'envoyer.



Remarque : la sortie Wireshark est identique sur les deux rayons.

Fonctionnalité de suivi des paquets Datapath Cisco IOS® XE

La fonctionnalité de suivi des paquets de chemin de données de Cisco IOS XE est utilisée pour analyser la manière dont le périphérique traite le trafic. Pour le configurer, vous devez créer une liste de contrôle d'accès incluant le trafic que vous souhaitez capturer sur les deux flux de trafic (entrant et sortant).

Dans ce scénario, les adresses IP NBMA sont utilisées.

```
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
```

Configurez ensuite la fonctionnalité fia-trace et définissez la condition de débogage pour utiliser la liste de contrôle d'accès. Enfin, commencez la condition.

```
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 : active le suivi fia détaillé, en l'arrêtant une fois la quantité de paquets configurés capturée
- debug platform condition ipv4 access-list <ACL-NAME> both : définit une condition sur le périphérique à l'aide de la liste d'accès précédemment configurée
- debug platform condition start : démarre la condition

Pour examiner le résultat de fia-trace, utilisez les commandes suivantes.

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

Résultat de la commande 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
```

Dans la sortie show platform packet-trace statistics, vous pouvez voir les compteurs pour les paquets traités par le périphérique. Cela vous permet de voir les paquets entrants et sortants, et de vérifier si le périphérique abandonne des paquets, ainsi que la raison de l'abandon.

Dans le résultat affiché, Spoke1 abandonne certains paquets avec la description Ipv4Acl. Pour analyser plus en détail ces paquets, la commande show platform packet-trace summary peut être utilisée.

Sortie du résumé 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
```

Avec cette sortie, vous pouvez voir chaque paquet arrivant et quittant le périphérique, ainsi que les interfaces d'entrée et de sortie. L'état du paquet est également affiché, indiquant s'il a été transféré, abandonné ou traité en interne (punt).

Dans cet exemple, cette sortie a aidé à identifier les paquets abandonnés par le périphérique. À l'aide de la commande show platform packet-trace packet <PACKET_NUMBER>, vous pouvez voir comment le périphérique traite ce paquet spécifique.

Sortie de Spoke1 show platform packet-trace packet <PACKET_NUMBER> :

<#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>

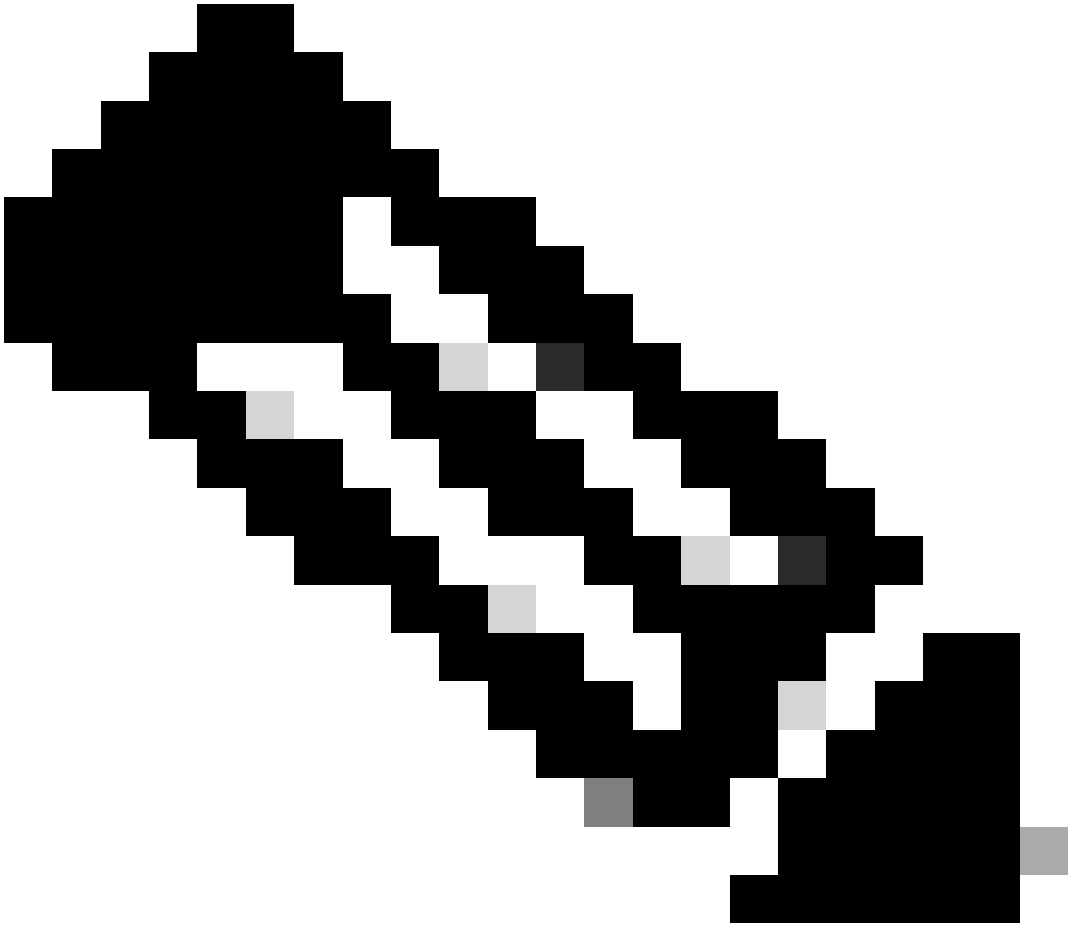
Lapsed time : 174 ns
Feature: INPUT_DROP

```
     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
```

Dans la première partie, vous pouvez voir l'interface d'entrée et de sortie, ainsi que l'état du paquet. Elle est suivie de la deuxième partie du résultat, dans laquelle vous pouvez trouver les adresses IP source et de destination, ainsi que le protocole.

Chaque phase suivante montre comment le périphérique traite ce paquet particulier. Cela permet d'avoir un aperçu de toutes les configurations telles que la traduction d'adresses de réseau (NAT) ou la liste d'accès ou d'autres facteurs qui pourraient avoir un impact sur elle.

Dans ce cas, on peut identifier que le protocole du paquet est ESP, l'IP source est l'adresse IP NBMA de Spoke2 et l'IP de destination est l'adresse IP NBMA de Spoke1. Cela indique qu'il s'agit du paquet manquant dans la négociation NHRP. En outre, il est observé qu'aucune interface de sortie n'est spécifiée dans une phase quelconque, ce qui suggère qu'un élément a affecté le trafic avant qu'il ne puisse être transféré. Lors de l'avant-dernière phase, vous pouvez voir que le périphérique abandonne le trafic entrant sur l'interface spécifiée (GigabitEthernet1). La dernière phase montre une liste d'accès d'entrée, suggérant qu'il peut y avoir une certaine configuration sur l'interface qui cause la perte.



Remarque : si, après avoir utilisé tous les outils de dépannage répertoriés dans ce document, les rayons impliqués dans la négociation ne montrent aucun signe d'abandon ou d'impact sur le trafic, le dépannage sur ces périphériques est terminé.

L’étape suivante doit consister à vérifier les périphériques intermédiaires entre eux, tels que les pare-feu, les commutateurs et les FAI.

Solution

Si un tel scénario est vu, l'étape suivante consiste à vérifier l'interface montrée dans les sorties précédentes. Cela implique de vérifier la configuration pour vérifier si quelque chose affecte le trafic.

Configuration de l’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
```

Dans le cadre de sa configuration, un groupe d'accès est appliqué à l'interface. Il est important de vérifier que les hôtes configurés sur la liste d'accès n'interfèrent pas avec le trafic utilisé pour la négociation 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 deuxième instruction de la liste d'accès refuse la communication entre l'adresse IP NBMA de Spoke2 et l'adresse IP NBMA de Spoke1, provoquant la perte précédemment observée. Après avoir supprimé le groupe d'accès de l'interface, la communication entre les deux rayons est réussie :

```
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
```

Le tunnel IPSEC est activé et affiche maintenant les encapsulations et les décapsulations sur les deux périphériques :

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:
```

Maintenant, la table DMVPN de Spoke1 affiche le mappage correct sur les deux entrées :

```
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
```

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.