DMVPNフェーズ2スポークツースポークトンネルのトラブルシューティング

内容

はじめに

このドキュメントでは、フェーズ2スポーク間DMVPNトンネルが確立されない場合のトラブルシューティング方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- ダイナミックマルチポイントバーチャルプライベートネットワーク(DMVPN)
- IKE/IPSECプロトコル
- Next Hop Resolution Protocol（NHRP）

使用するコンポーネント

このドキュメントは、次のソフトウェアバージョンに基づいています。

- Cisco CSR1000V(VXE)：バージョン17.03.08

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このド

キュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

このドキュメントでは、一般的なDMVPNの問題に関するさまざまなトラブルシューティングツールを設定して使用する方法について説明します。 この問題はフェーズ2 DMVPNトンネルのネゴシエーションの失敗で、発信元スポークでは、DMVPN状態が正しい非ブロードキャストマルチアクセス(NBMA)/トンネルマッピングを使用して宛先スポークに表示されます。ただし、宛先スポークでは誤ったマッピングが表示されます。

理論的背景

DMVPNフェーズ2を設定する際には、スポーク間トンネルがどのように確立されるかを理解することが重要です。このセクションでは、このフェーズにおけるNHRPプロセスの理論上の概要を簡単に説明します。

DMVPNフェーズ2では、動的なスポーク間トンネルをオンデマンドで構築できます。 これが可能なのは、DMVPNクラウド内のすべてのデバイス（ハブおよびスポーク）で、トンネルインターフェイスのモードがGeneric Routing Encapsulation（GRE；総称ルーティングカプセル化）マルチポイントに変更されるためです。このフェーズの主な機能の1つは、他のデバイスがハブをネクストホップとして認識しないことです。代わりに、すべてのスポークに相互のルーティング情報があります。 フェーズ2でスポーク間トンネルを確立すると、スポークが他のスポークに関する情報を学習するNHRPプロセスがトリガーされ、NBMAとトンネルIPアドレス間でマッピングが行われます。

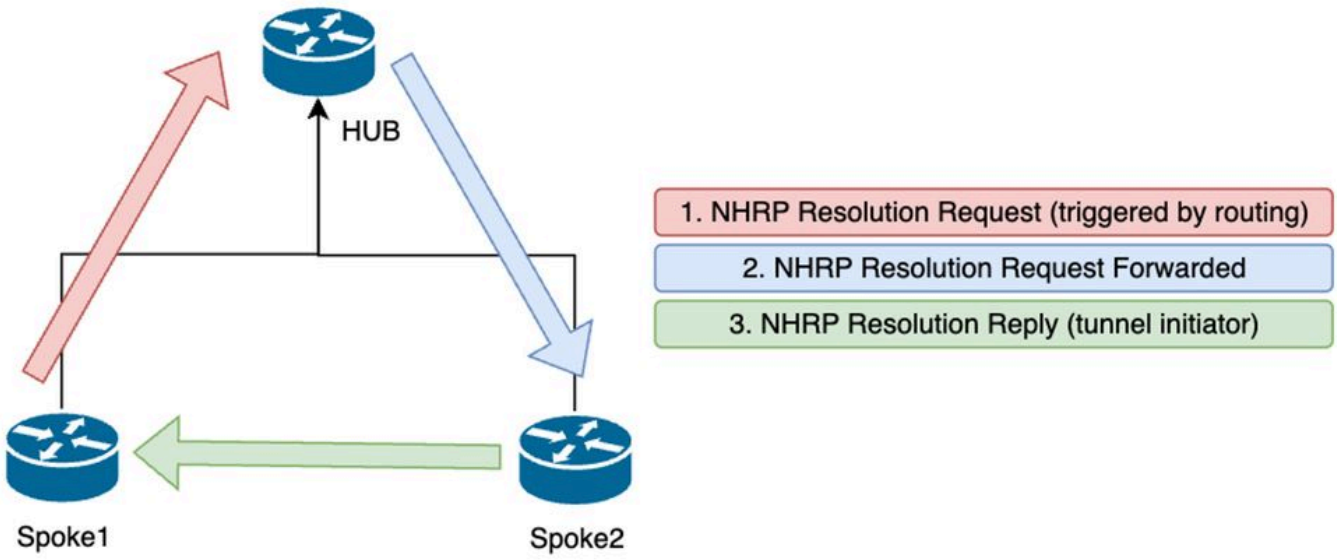
次のステップでは、NHRP解決プロセスがどのようにトリガーされるかを示します。

1. 送信元スポークが宛先スポークのLANに到達しようとすると、解決要求メッセージをトリガーするルートルックアップを実行して、宛先スポークのNBMAアドレスを取得します。送信元スポークは、この初期メッセージをハブに送信します。

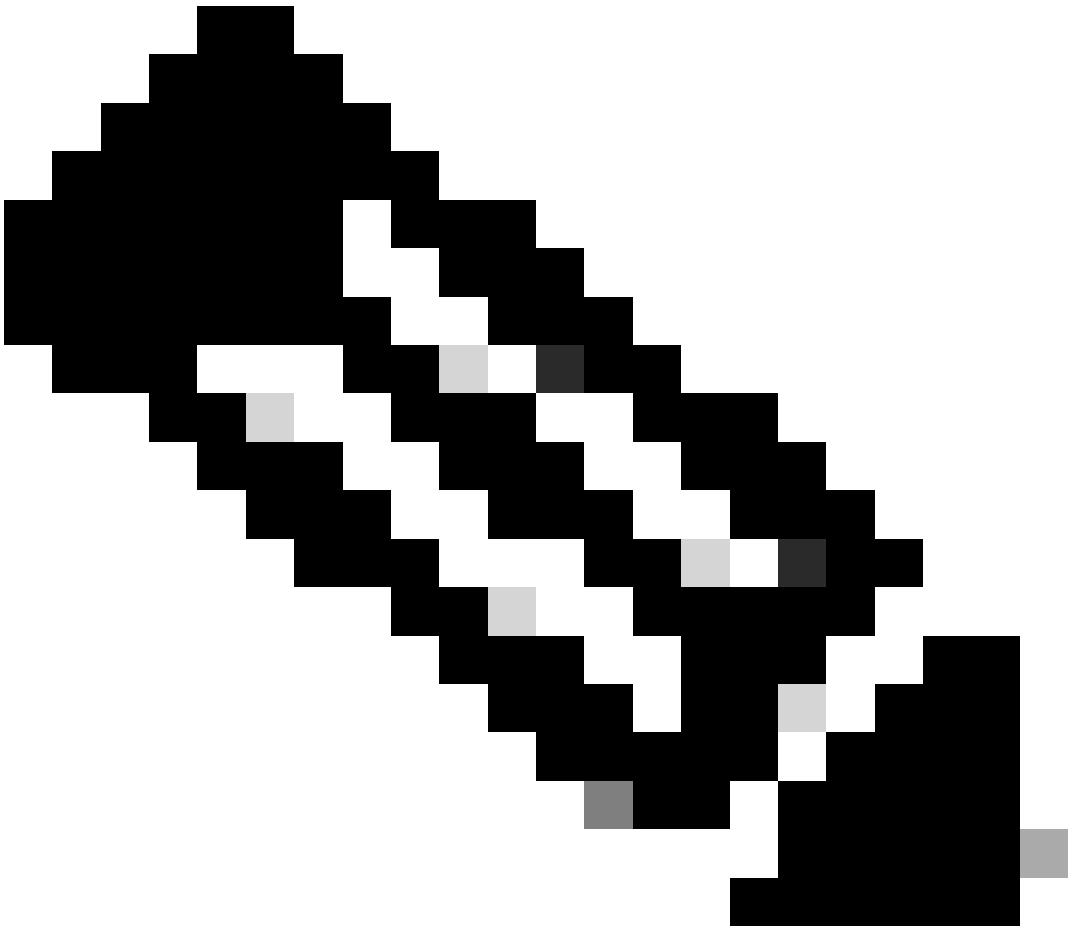
2. ハブは解決要求を受信し、宛先スポークに転送します。

3. 宛先スポークは、送信元スポークに解決応答を送信します。トンネル設定にリンクされたIPSECプロファイルがある場合：

    - NHRP解決プロセスは、IKE/IPSECプロトコルが確立できるまで遅延されます。
    - 宛先スポークがIKE/IPSECトンネルを開始して確立します。
    - 次に、NHRPプロセスが再開され、宛先スポークは転送方式としてIPSECトンネルを使用して、解決応答を送信元スポークに送信します。



フェーズ2のスポーク間のNHRPメッセージフロー

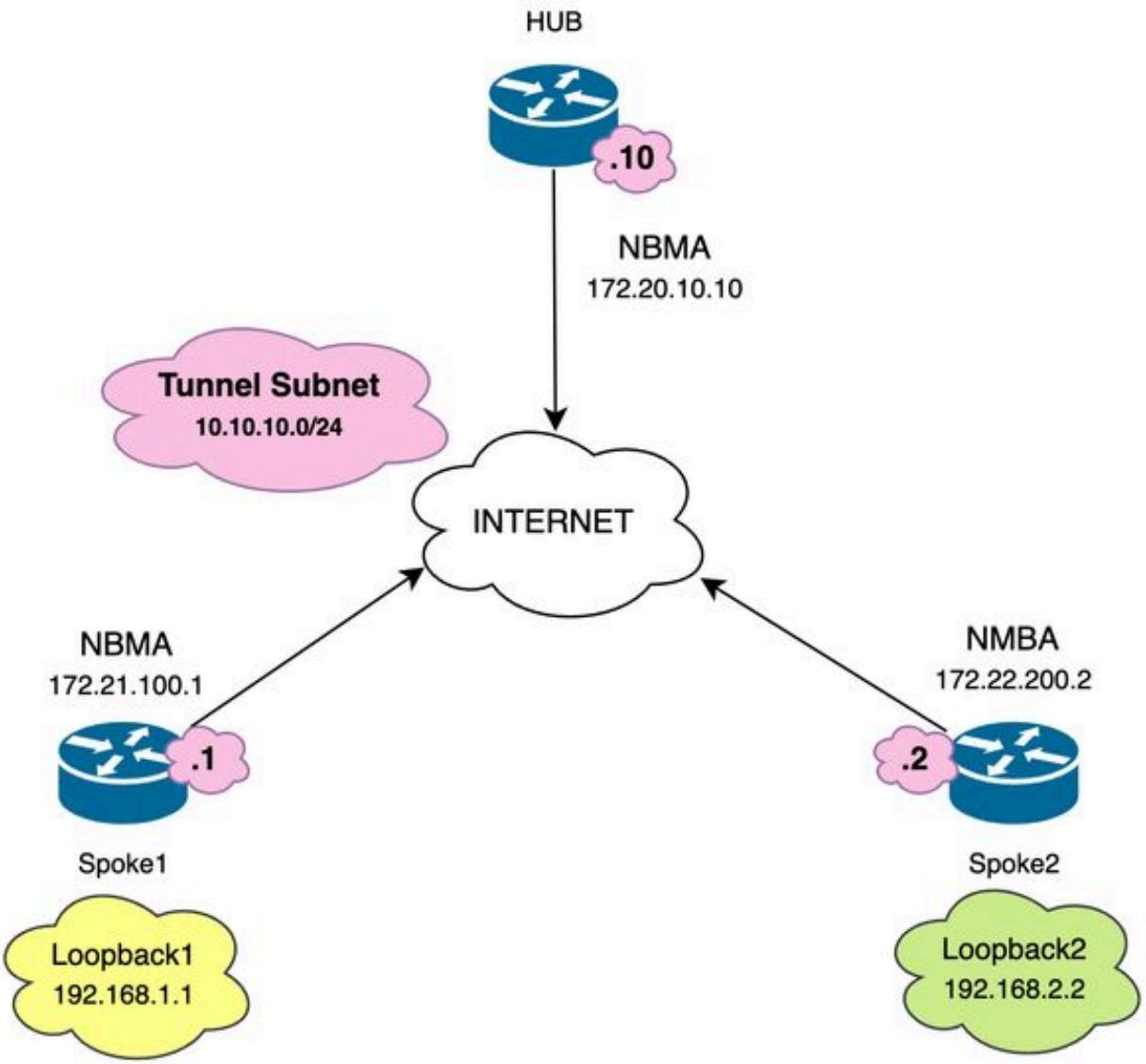


注：解決プロセスを開始する前に、すべてのスポークがすでにハブに登録されている必

要があります。

トポロジ

次の図は、このシナリオで使用するトポロジを示しています。



使用するネットワークダイアグラムとIPサブネット

トラブルシューティングの手順

このシナリオでは、Spoke1とSpoke2の間にスポーク間トンネルが確立されず、互いに到達できないため、（ループバックインターフェイスで表される）ローカルリソース間の通信に影響します。

```
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)
```

初期検証

このような状況が発生した場合は、まずトンネル設定を検証し、両方のデバイスに正しい値が設定されていることを確認することが重要です。トンネルの設定を確認するには、show running-config interface tunnel<ID>コマンドを実行します。

スポーク1のトンネル設定：

<#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
```

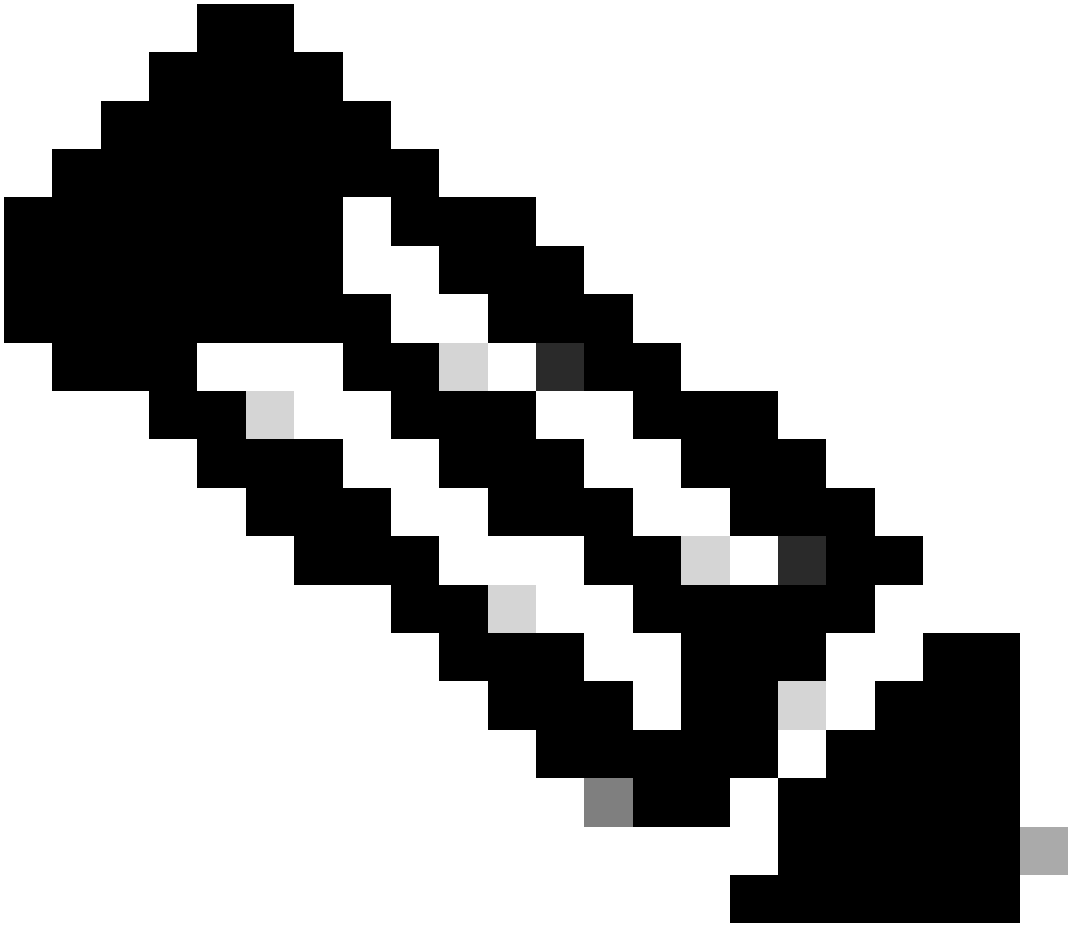
スポーク2のトンネル設定：

<#root>

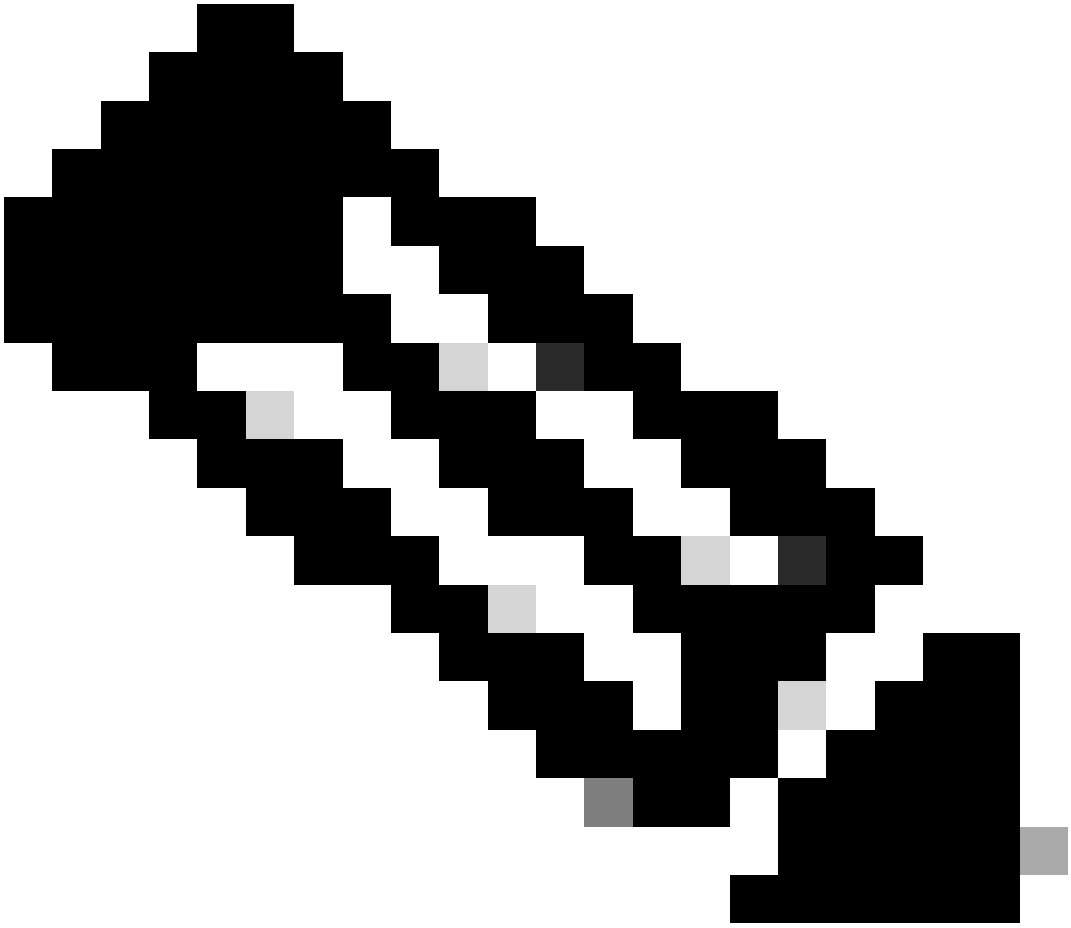
```
SPOKE2#show running-config interface tunnel10
Building configuration...

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
```

設定では、ハブへのマッピングが正しいこと、NHRP認証文字列がデバイス間で一致していること、両方のスポークに同じDMVPNフェーズが設定されていること、およびIPSEC保護を使用している場合は正しい暗号設定が適用されていることを検証する必要があります。

設定が正しく、IPSEC保護が含まれている場合は、IKEプロトコルとIPSECプロトコルが正しく動作していることを確認する必要があります。これは、NHRPが完全にネゴシエートするための転送方式としてIPSECトンネルを使用するためです。IKE/IPSECプロトコルの状態を確認するには、コマンドshow crypto IPSEC sa peer x.x.x.x（x.x.x.xは、トンネルを確立しようとしているスポークのNBMA IPアドレス）を実行します。



注:IPSECトンネルがアップしているかどうかを確認するには、着信および発信カプセル化セキュリティペイロード(ESP)セクションにトンネル情報（SPI、トランスフォームセットなど）が含まれている必要があります。このセクションに示されているすべての値は、両端で一致する必要があります。



注:IKE/IPSECに関する問題が特定された場合は、それらのプロトコルを対象としたトラブルシューティングを行う必要があります。

Spoke1上のIKE/IPSECトンネルステータス：

<#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:

Spoke2のIKE/IPSECトンネル状態：

<#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:
```

出力は、両方のスポークでIPSECトンネルが稼働しているが、Spoke2では暗号化されたパケット（カプセル化）を示しているが、復号化されたパケット（カプセル化解除）は示していないことを示しています。一方、Spoke1はIPSECトンネルを通過するパケットを示していません。これは、NHRPプロトコルに問題がある可能性があることを示しています。

トラブルシューティング ツール

初期検証を実行し、設定とIKE/IPSECプロトコル（必要な場合）を裏付けても通信の問題が発生しない場合は、この項で説明するツールを使用してトラブルシューティングを続行できます。

便利なコマンド

show dmvpn interface tunnel<ID>コマンドにより、DMVPN固有のセッション情報(NBMA/トンネルIPアドレス、トンネルの状態、アップ/ダウン時間、および属性)が得られます。detailキーワードを使用すると、暗号化セッションまたは暗号化ソケットの詳細情報を表示できます。トンネルの状態が両端で一致する必要があることを説明することが重要です。

スポーク1 show dmvpn interface tunnel<ID>の出力：

```
<#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
```

スポーク2のshow dmvpn interface tunnel<ID> の出力：

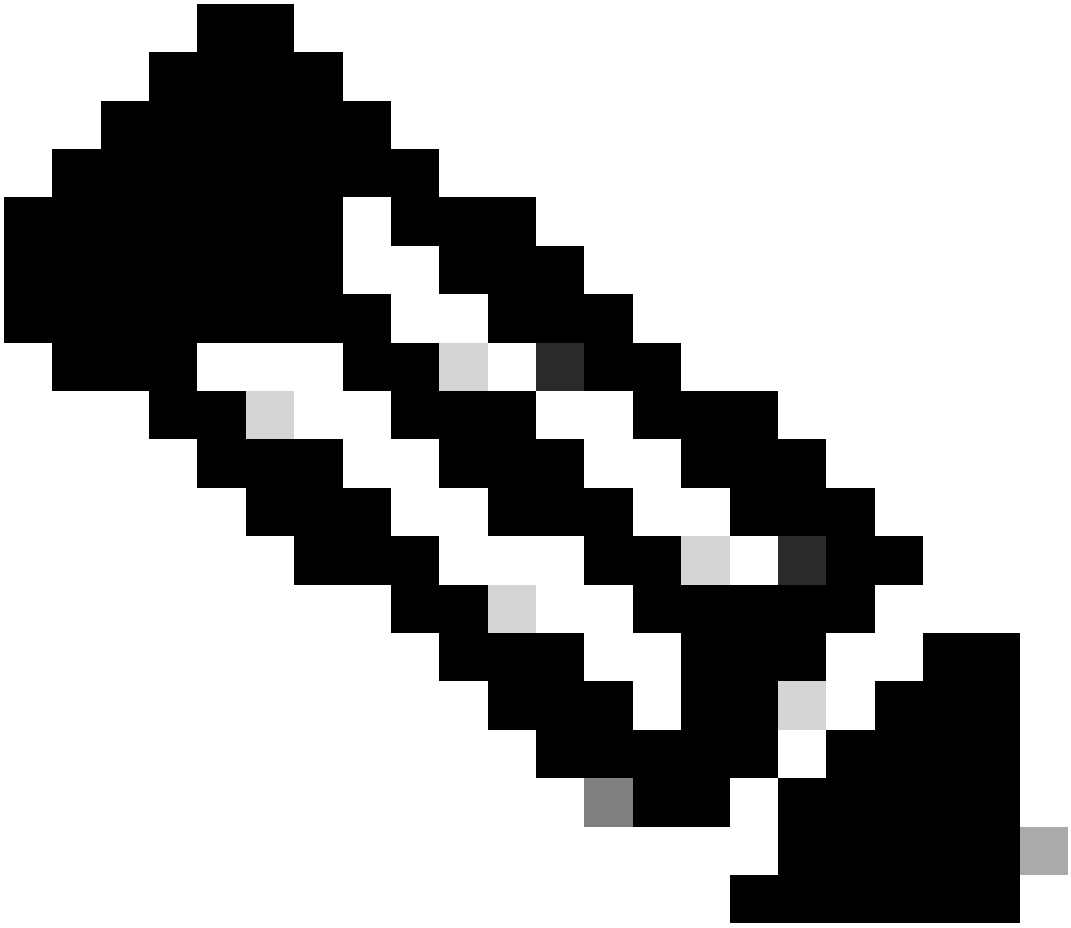
<#root>

```
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
```

各デバイスの出力には、スポークごとに異なる情報が表示されます。Spoke1テーブルでは、Spoke2のエントリに正しいNBMA IPアドレスが含まれておらず、属性が不完全である(I2)ことがわかります。一方、Spoke2テーブルには正しいマッピング（NBMA/トンネルIPアドレス）と、トンネルが完全にネゴシエートされたことを示すup状態が表示されます。

トラブルシューティングプロセスでは、次のコマンドが役に立つ場合があります。

- show ip nhrp:NHRPマッピング情報を表示します。
- show ip nhrp traffic interface tunnel10:NHRPトラフィック統計情報を表示します

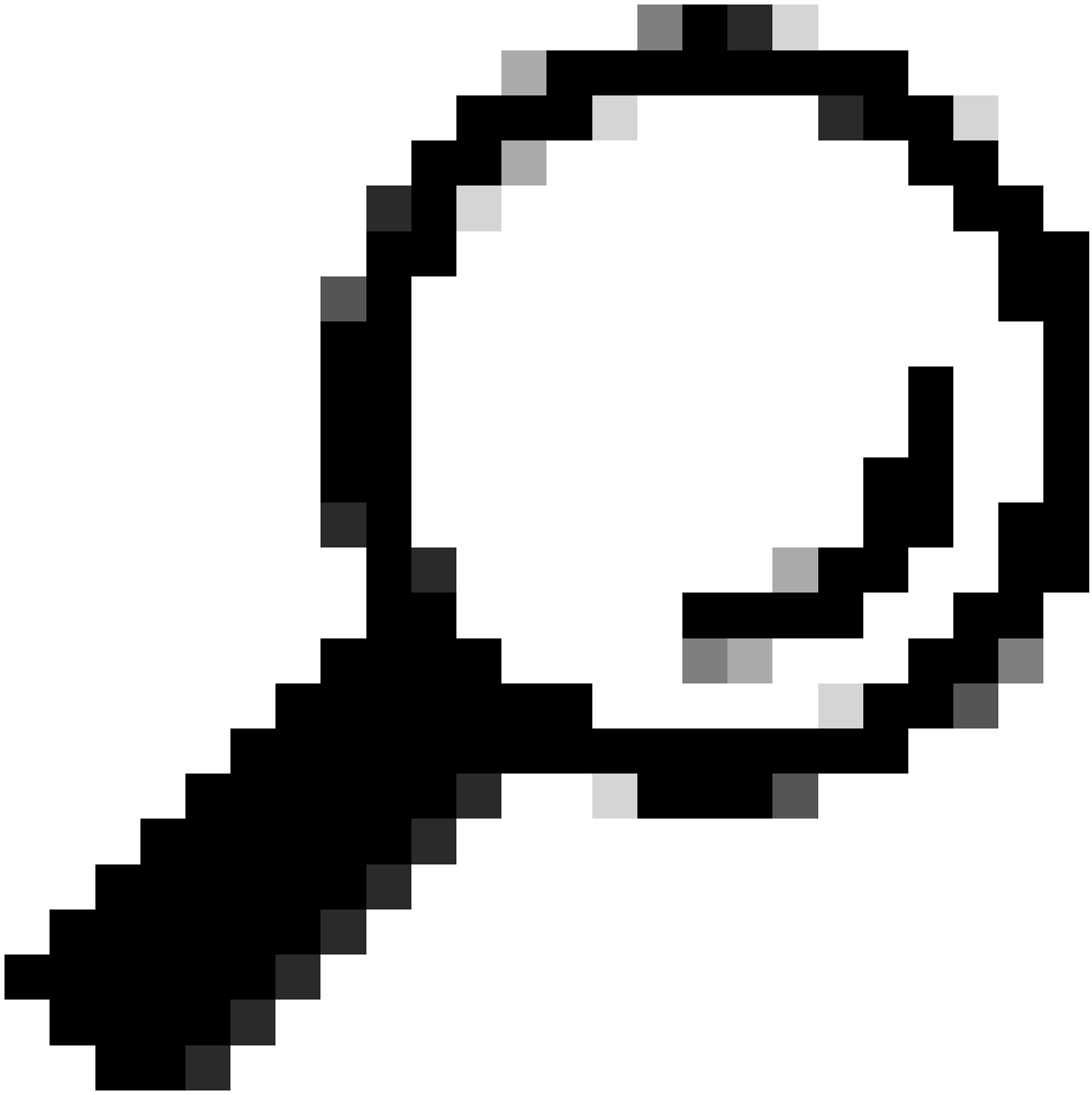


注：コマンドの仕様（構文、説明、キーワードなど）については、「コマンドリファレンス：Cisco IOSセキュリティコマンドリファレンス：コマンドSからZ」を参照してください。

デバッグ

上記の情報を確認し、トンネルにネゴシエーションの問題が発生していることを確認したら、NHRPパケットの交換方法を観察するためにデバッグを有効にする必要があります。次のデバッグは、関連するすべてのデバイスで有効にする必要があります。

1. debug dmvpn condition peer NBMA x.x.x.x（x.x.x.xはリモートデバイスのIPアドレス）
2. debug dmvpn all：このコマンドはISAKMP、IKEv2、IPSEC、DMVPN、およびNHRPデバッグコマンドを有効にします。



ヒント：デバッグを有効にするたびにpeer conditionコマンドを使用して、その特定のトンネルのネゴシエーションを確認できるようにすることをお勧めします。

完全なNHRPフローを確認するために、各デバイスで次のdebugコマンドを使用しました。

スポーク1

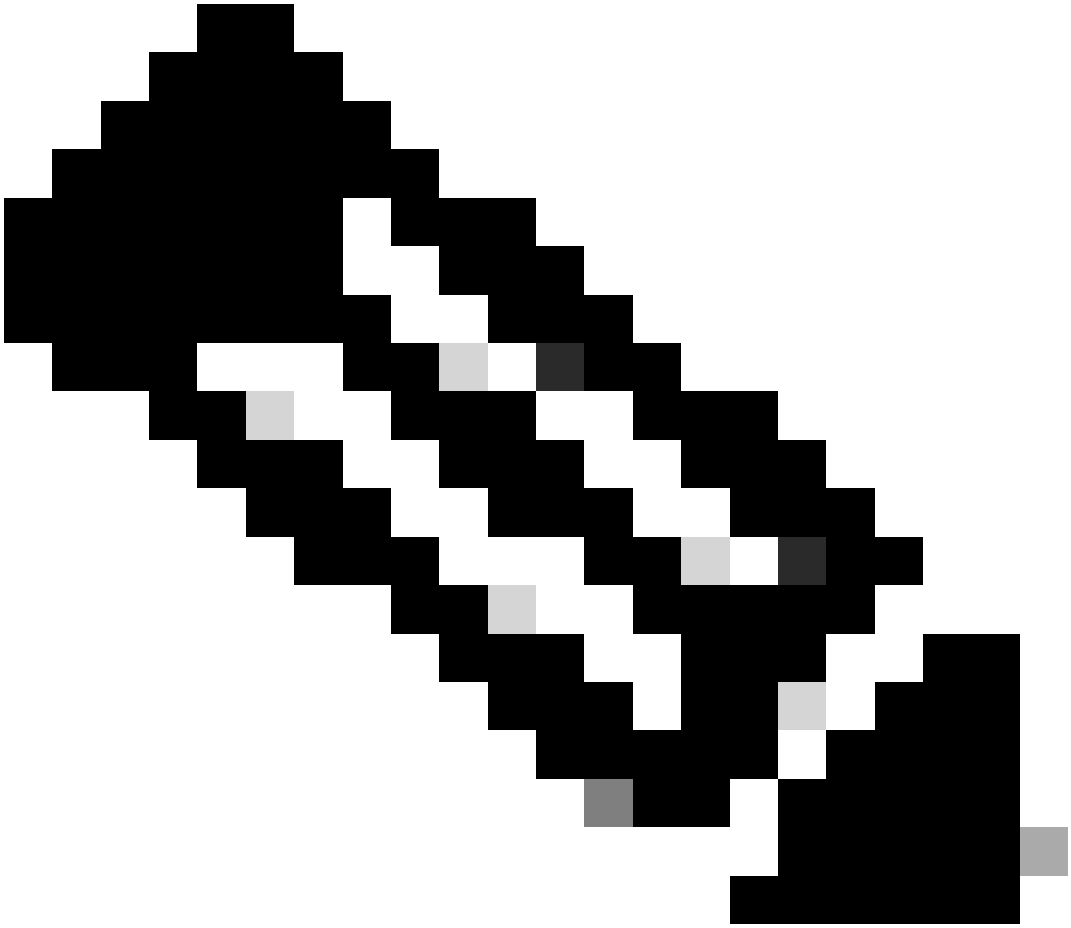
```
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```

ハブ

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

スポーク2

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



注：デバッグは、関連するすべてのデバイスで同時に有効にして収集する必要があります。

すべてのデバイスで有効にされているデバッグは、show debugコマンドで表示されます。

```
<#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
```

すべてのデバッグを収集した後、送信元スポーク(Spoke1)でデバッグの分析を開始する必要があります。これにより、ネゴシエーションを最初からトレースできます。

Spoke1のデバッグ出力：

<#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

Spoke1 NHRPプロセスが開始されると、ログにはデバイスがNHRP解決要求を送信していることが示されます。パケットには、送信元スポーク(Spoke1)のNBMA IPアドレスおよびトンネルIPアドレスである、src NMBAやsrc protocolなどの重要な情報があります。宛先スポーク(Spoke2)のトンネルIPアドレスを持つdst protocol値（スポーク1のIPアドレス）も確認できます。これは、Spoke1がマッピングを完了するためにSpoke2のNBMAアドレスを要求していることを示しています。 また、パケット上で、パスに沿ってパケットを追跡するのに役立つreqid値を見つけることができます。この値はプロセス全体を通じて同じままであり、NHRPネゴシエーションの特定のフローを追跡するのに役立つ可能性があります。パケットには、NHRP認証文字列など、ネゴシエーションに重要な別の値があります。

デバイスがNHRP解決要求を送信すると、ログには再送信が送信されたことが示されます。これは、デバイスがNHRP解決応答を受信していないためにパケットを再送信しているためです。Spoke1は応答を認識していないため、パス内の次のデバイス(HUB)でそのパケットを追跡する必要があります。

ハブのデバッグ出力：

<#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
```

reqidの値を使用して、HUBがSpoke1によって送信された解決要求を受信することを確認できます。このパケットでは、src NBMAおよびsrc protocolの値はSpoke1からの情報であり、dst protocolの値はSpoke1のデバッグで確認されたようにSpoke2のトンネルIPです。ハブは解決要求を受信すると、ルートルックアップを実行し、パケットをSpoke2に転送します。転送されたパケット内に、ハブは自身の情報（NBMA IPアドレスおよびトンネルIPアドレス）を含む内線番号を追加します。

上記のデバッグは、ハブがスポーク2に解決要求を正しく転送していることを示しています。したがって、次のステップでは、Spoke2がそれを受信し、正しく処理し、解決応答をSpoke1に送信していることを確認します。

Spoke2デバッグ出力：

<#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
```

reqidは前の出力に表示される値と一致するため、これによりSpoke1から送信されたNHRP解決要求パケットがSpoke2に到達することが確認されました。このパケットによってSpoke2でルートルックアップがトリガーされ、解決要求が自分宛てであることが認識されるため、Spoke2はSpoke1の情報をNHRPテーブルに追加します。解決応答パケットをSpoke1に返信する前に、デバイスは自身の情報（NBMA IPアドレスおよびトンネルIPアドレス）を追加します。これにより、Spoke1はそのパケットを使用してデータベースに情報を追加できます。

すべてのデバッグから判断すると、Spoke2から送信されたNHRP解決応答はSpoke1に到達していません。HUBは、NHRP解決要求パケットを期待どおりに受信して転送するため、問題から廃棄される可能性があります。したがって、次の手順では、Spoke1とSpoke2の間でキャプチャを取得して、問題の詳細を取得します。

Embedded Packet Capture

組み込みパケットキャプチャ機能を使用すると、デバイスを通過するトラフィックを分析できま

す。これを設定する最初の手順は、両方のトラフィックフロー（着信と発信）でキャプチャするトラフィックを含むアクセスリストを作成することです。

このシナリオでは、NBMA IPアドレスが使用されます。

```
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
```

次に、monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 interface <WAN_INTERFACE> both コマンドを使用してキャプチャを設定し、monitor capture <CAPTURE_NAME> startコマンドを使用してキャプチャを開始します。

Spoke1とSpoke2で設定をキャプチャします。

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

キャプチャの出力を表示するには、show monitor capture <CAPTURE_NAME> buffer briefコマンドを使用します。

出力Spoke1をキャプチャします。

<#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

出力Spoke2をキャプチャします。

<#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

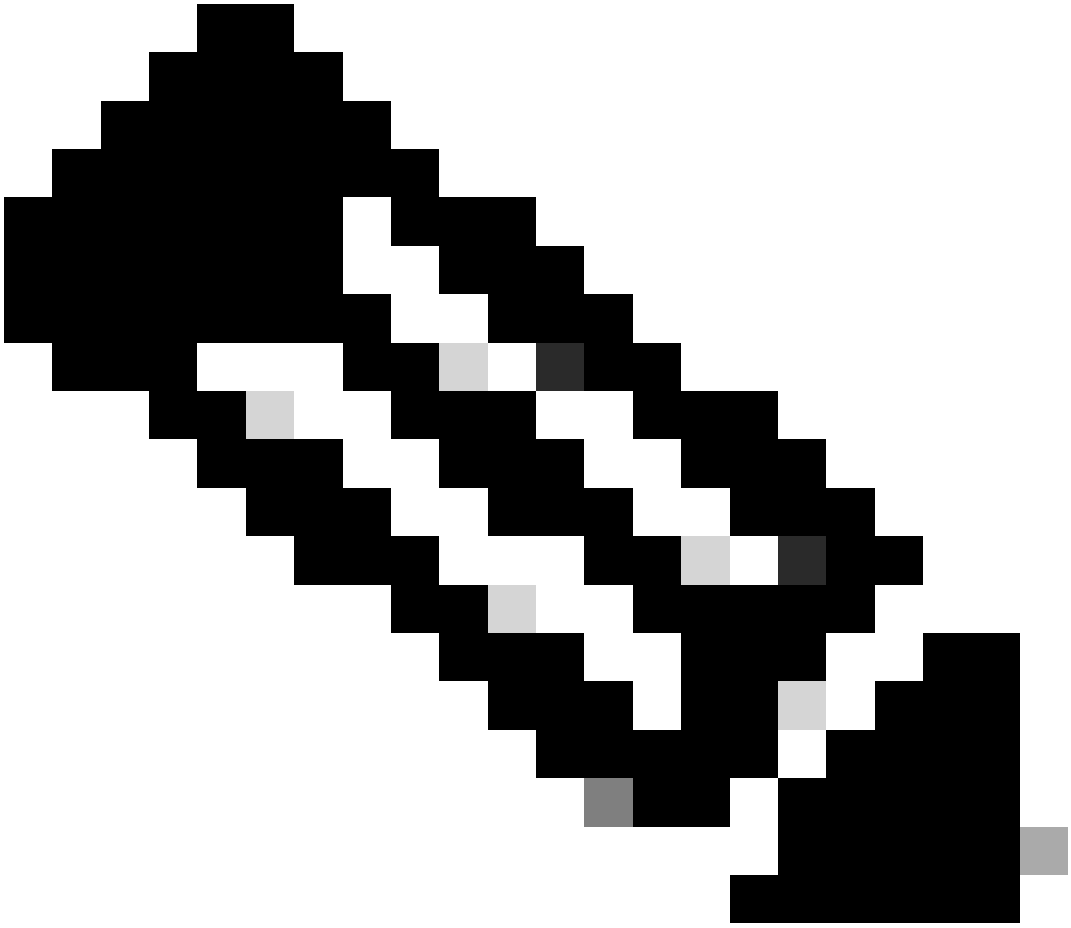
10 70 0.108988 172.21.100.1 -> 172.22.200.2 0 BE ICMP

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
```

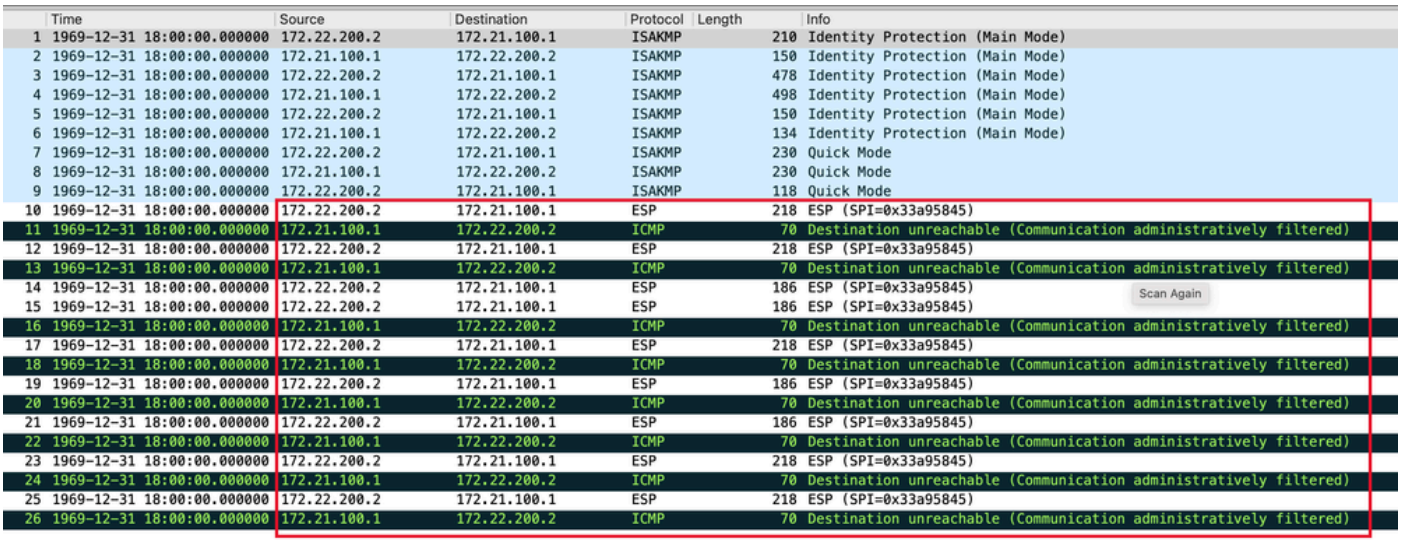
キャプチャの出力は、初期パケットがIKE/IPSECネゴシエーションを示すUDPトラフィックであることを示しています。その後、Spoke2は解決応答をSpoke1に送信し、これはESPトラフィック（パケット9）と見なされます。この後、予想されるトラフィックフローはESPですが、次に確認されるパケットは、Spoke1からSpoke2へのICMPトラフィックです。

パケットをより詳細に分析するには、show monitor capture <CAPTURE_NAME> buffer dumpコマンドを実行して、デバイスからpcapファイルをエクスポートできます。次に、デコーダツールを使用してダンプ出力をpcapファイルに変換し、Wiresharkで開くことができます。



注：シスコのパケットアナライザには、キャプチャの設定、例、およびデコーダが用意されています(Cisco TACツール：Packet Capture Config Generator and Analyzer)

Wiresharkの出力：

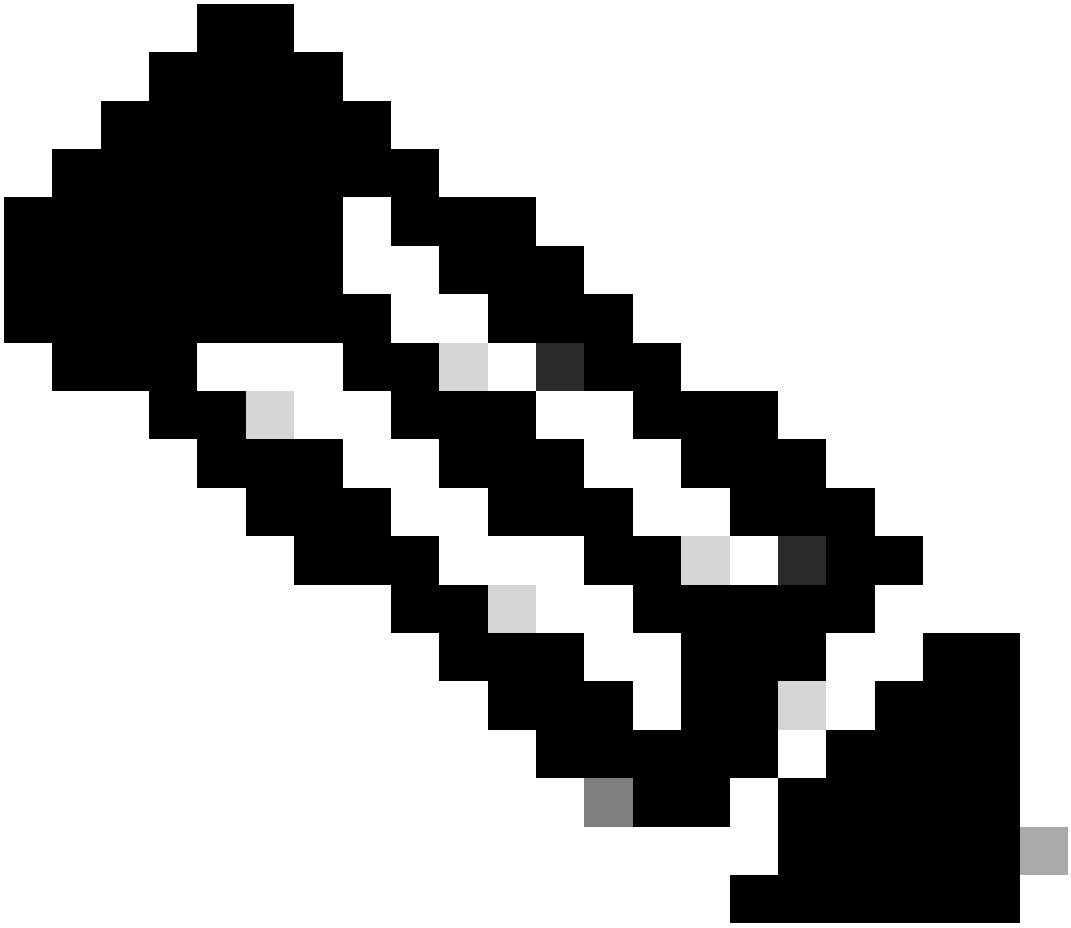


	Time	Source	Destination	Protocol	Length	Info
1	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	210	Identity Protection (Main Mode)
2	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	150	Identity Protection (Main Mode)
3	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP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

Wiresharkでの出力のキャプチャ

ICMPパケットの内容に「Destination unreachable (Communication administratively filtered)」というエラーメッセージが表示される。これは、ルータACLやファイアウォールなど、パス上のトラフィックに影響を与える何らかのフィルタがあることを示します。ほとんどの場合、フィルタはパケットを送信するデバイス（この場合はSpoke1）で設定されますが、中間のデバイスでも同様に送信できます。



注：Wiresharkの出力は、両方のスポークで同じです。

Cisco IOS® XEデータパスパケットトレース機能

Cisco IOS XEデータパスパケットトレース機能は、デバイスがトラフィックを処理する方法を分析するために使用されます。これを設定するには、両方のトラフィックフロー（着信と発信）でキャプチャするトラフィックを含むアクセスリストを作成する必要があります。

このシナリオでは、NBMA IPアドレスが使用されます。

```
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
```

次に、fia-trace機能を設定し、アクセスリストを使用するようにデバッグ条件を設定します。最

後に、条件を開始します。

```
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:詳細なfiaトレースを有効にし、設定されたパケットの量をキャプチャしてから停止します
- debug platform condition ipv4 access-list <ACL-NAME> both:以前に設定したアクセスリストを使用して、デバイスの条件を設定します。
- debug platform condition start:条件を開始します。

fia-traceの出力を確認するには、次のコマンドを使用します。

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

Spoke1 show platform packet-trace statisticsの出力：

<#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
```

show platform packet-trace statisticsの出力では、デバイスで処理されたパケットのカウンタを確認できます。これにより、着信パケットと発信パケットを確認し、デバイスがパケットをドロップしているかどうかを、ドロップの理由とともに確認できます。

示されている出力では、Spoke1が説明Ipv4Aclを含むいくつかのパケットをドロップしています。これらのパケットをさらに分析するには、コマンドshow platform packet-trace summaryを使用できます。

Spoke1 show platform packet-trace summary の出力：

<#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
```

この出力から、デバイスを出入りする各パケットと、入力および出力インターフェイスを確認できます。パケットのステータスも表示され、パケットが転送(forwarded)、廃棄(dropped)、または内部処理(punt)されたかどうかが示されます。

この例では、この出力はデバイスによってドロップされているパケットを特定するのに役立ちます。show platform packet-trace packet <PACKET_NUMBER>コマンドを使用すると、デバイスが特定のパケットを処理する方法を確認できます。

Spoke1 show platform packet-trace packet <PACKET_NUMBER> の出力：

<#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

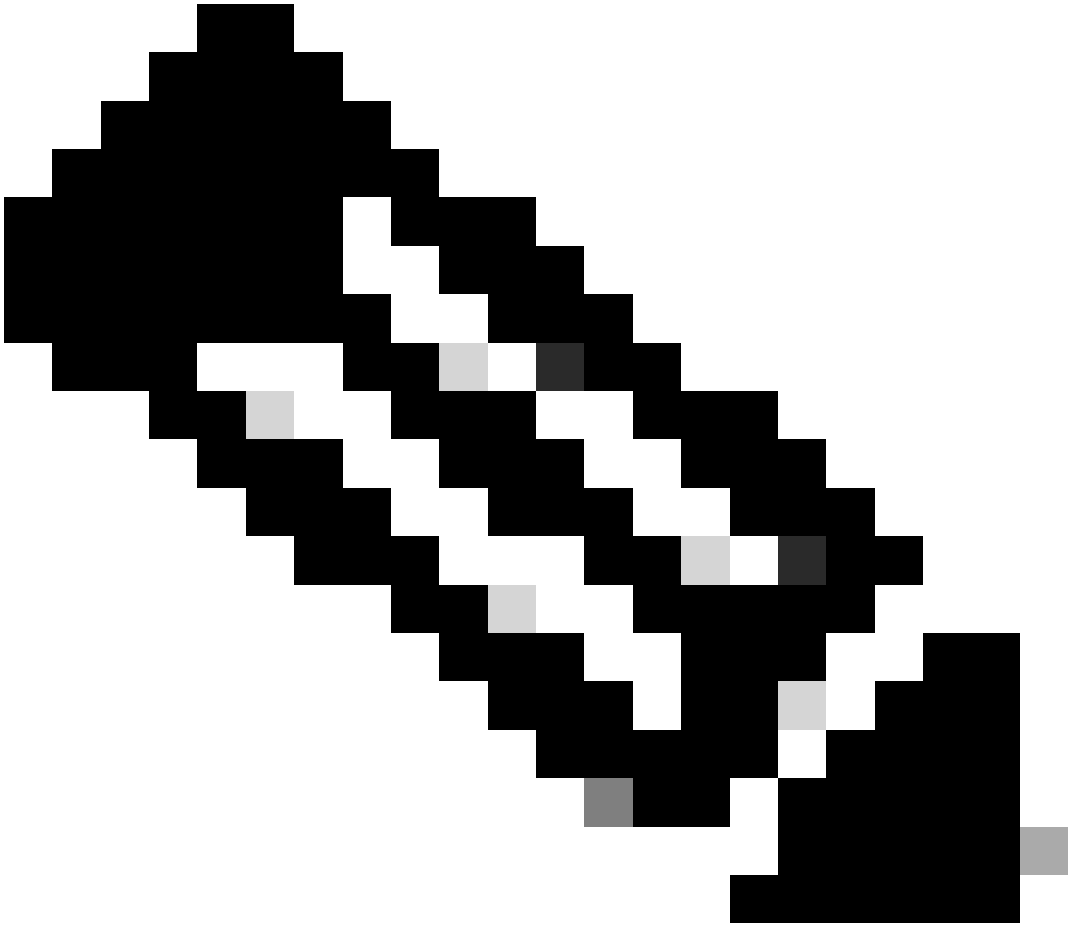
Entry : Input - 0x8126e568

```
Input : GigabitEthernet1

     Output : <unknown>

     Lapsed time : 116 ns

Feature: IPV4_INPUT_ACL

     Entry : Input - 0x81271f70

Input : GigabitEthernet1

     Output : <unknown>

     Lapsed time : 12915 ns
```

最初の部分では、入力インターフェイスと出力インターフェイス、およびパケットの状態を確認できます。次に、出力の2番目の部分に続いて、送信元と宛先のIPアドレスとプロトコルを確認できます。

後続の各フェーズでは、デバイスがこの特定のパケットを処理する方法が示されます。 これにより、ネットワークアドレス変換(NAT)やアクセスリストなどの設定、またはそれに影響を与える可能性のあるその他の要因に関する洞察が得られます。

この場合、パケットのプロトコルはESP、送信元IPはSpoke2のNBMA IPアドレス、宛先IPはSpoke1のNBMA IPアドレスであることが確認できます。これは、NHRPネゴシエーションでパケットが欠落していることを示します。また、どのフェーズでも出力インターフェイスが指定されていないことがわかり、トラフィックが転送される前に何らかの影響を受けていることが示唆されます。Penultimateフェーズでは、デバイスが指定されたインターフェイス(GigabitEthernet1)で着信トラフィックをドロップしていることを確認できます。最後のフェーズは入力アクセスリストを示し、廃棄を引き起こしているインターフェイス上に何らかの設定が存在する可能性を示唆しています。



注：このドキュメントに記載されているすべてのトラブルシューティングツールを使用した後で、ネゴシエーションに関係するスポークからトラフィックがドロップまたは影響を受けていることを示す兆候が見られない場合は、これらのデバイスでのトラブルシューティングを終了します。

次のステップでは、ファイアウォール、スイッチ、ISPなど、中間のデバイスを確認する必要があります。

解決方法

このようなシナリオが見られる場合、次のステップは前の出力に示されているインターフェイスを確認することです。 これには、設定をチェックして、トラフィックに影響を与える何かがあるかどうかを確認する作業が含まれます。

WANインターフェイス設定：

<#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
```

設定の一部として、インターフェイスにaccess-groupが適用されています。アクセスリストで設定されているホストが、NHRPネゴシエーションに使用されるトラフィックに干渉していないことを確認することが重要です。

<#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)
```

access-listの2番目の文がSpoke2のNBMA IPアドレスとSpoke1のNBMA IPアドレス間の通信を拒否しているため、前述のドロップが発生します。インターフェイスからaccess-groupを削除すると、2つのスポーク間の通信は成功します。

```
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
```

IPSECトンネルはアップ状態で、両方のデバイスでカプセル化とカプセル化解除が表示されています。

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:
```

Spoke1のDMVPNテーブルは両方のエントリで正しいマッピングを示しています。

<#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
```

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。