Fehlerbehebung: DMVPN Phase 2 Spoke-to-Spoke-Tunnel

Inhalt

Einleitung

In diesem Dokument wird die Fehlerbehebung für einen Phase-2-Spoke-to-Spoke-DMVPN-Tunnel beschrieben, wenn dieser nicht eingerichtet wird.

Voraussetzungen

Anforderungen

Cisco empfiehlt, sich mit den folgenden Themen vertraut zu machen:

- Dynamic Multipoint Virtual Private Network (DMVPN)
- IKE/IPSEC-Protokolle
- Next Hop Resolution Protocol (NHRP)

Verwendete Komponenten

Dieses Dokument basiert auf der folgenden Softwareversion:

- Cisco CSR1000V (VXE) - Version 17.03.08

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten

Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

In diesem Dokument wird beschrieben, wie bei einem allgemeinen DMVPN-Problem verschiedene Tools zur Fehlerbehebung konfiguriert und verwendet werden. Das Problem besteht darin, dass ein Phase-2-DMVPN-Tunnel nicht ausgehandelt werden konnte, in dem die Quelle gesprochen hat. Der DMVPN-Status wird "UP" mit der korrekten NBMA- (Non-Broadcast Multi-Access)/Tunnel-Zuordnung zum Ziel-Spoke angezeigt. Auf der Ziel-Spoke wird jedoch eine falsche Zuordnung angezeigt.

Theoretischer Hintergrund

Es ist wichtig zu wissen, wie Spoke-to-Spoke-Tunnel eingerichtet werden, wenn eine DMVPN Phase 2 eingerichtet ist. Dieser Abschnitt bietet eine kurze theoretische Zusammenfassung des NHRP-Prozesses während dieser Phase.

In DMVPN Phase 2 können dynamische Spoke-to-Spoke-Tunnel nach Bedarf erstellt werden. Dies ist möglich, da sich der Modus der Tunnelschnittstelle auf allen Geräten in der DMVPN-Cloud (Hub und Stationen) in Generic Routing Encapsulation (GRE) Multipoint ändert. Eines der wichtigsten Merkmale dieser Phase ist, dass der Hub von den anderen Geräten nicht als Next-Hop wahrgenommen wird. Stattdessen haben alle Stationen die Routing-Informationen voneinander. Bei der Einrichtung eines Spoke-to-Spoke-Tunnels in Phase 2 wird ein NHRP-Prozess ausgelöst, bei dem die Stationen die Informationen über andere Stationen abrufen und eine Zuordnung zwischen NBMA- und Tunnel-IP-Adressen vornehmen.

In den nächsten Schritten wird aufgelistet, wie der NHRP-Auflösungsprozess ausgelöst wird:

1. Wenn die Source-Spoke versucht, das LAN der Ziel-Spoke zu erreichen, führt sie eine Routensuche durch, die die Meldung zur Auflösungsanforderung auslöst, um die NBMA-Adresse der Ziel-Spoke zu erhalten. Die Quelle sprach und sendete diese erste Nachricht an den Hub.

2. Der Hub empfängt die Lösungsanforderung und leitet sie an die Ziel-Spoke weiter.

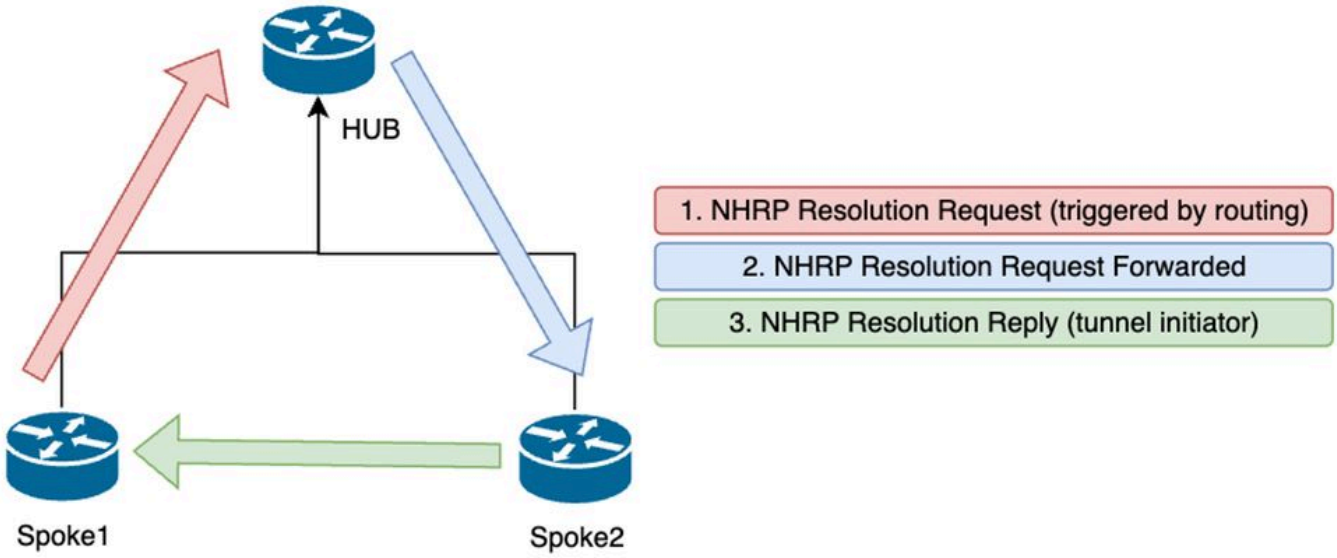
3. Die Ziel-Spoke sendet die Auflösungsantwort an die Quell-Spoke. Wenn mit der Tunnelkonfiguration ein IPSEC-Profil verknüpft ist:

    - Der NHRP-Auflösungsprozess wird verzögert, bis die IKE/IPSEC-Protokolle eingerichtet werden können.

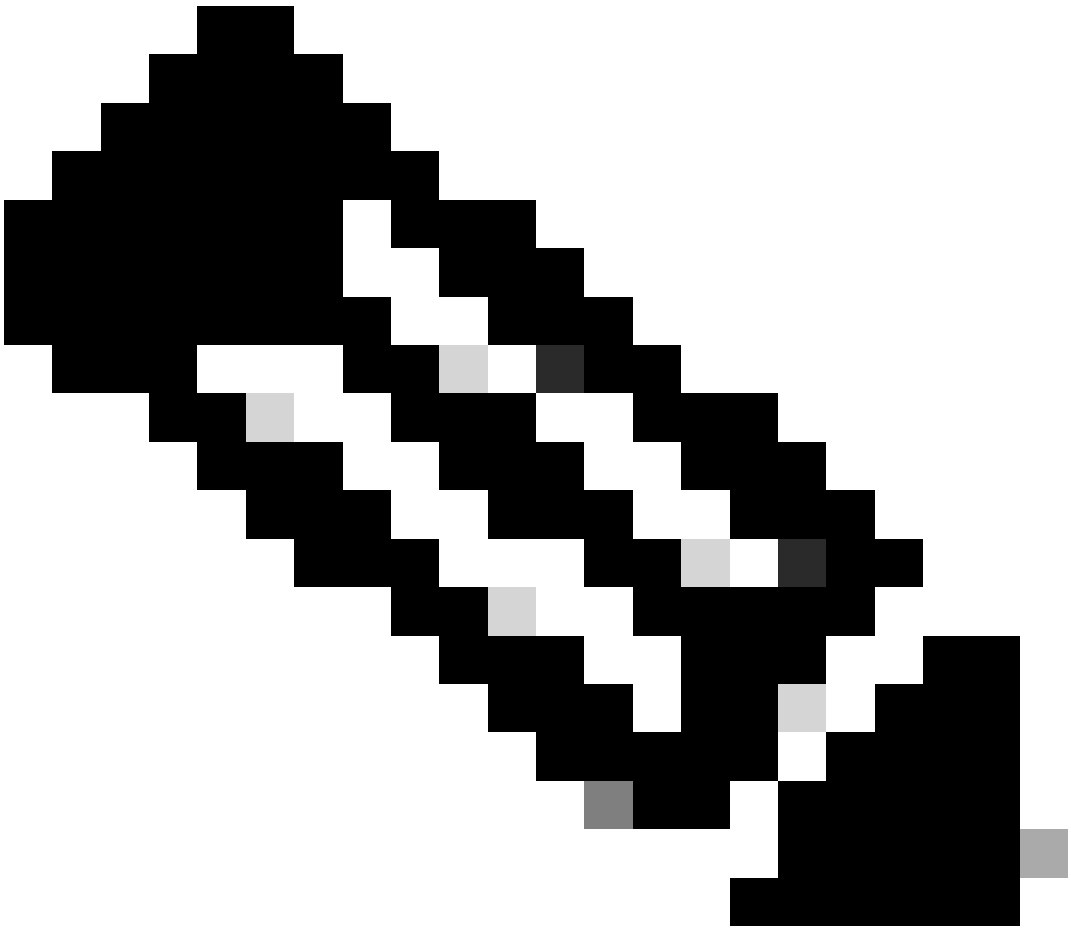
    - Die Ziel-Spoke initiiert und erstellt die IKE/IPSEC-Tunnel.

    - Anschließend wird der NHRP-Prozess wieder aufgenommen, und die Ziel-Spoke sendet mithilfe des IPSEC-Tunnels als Transportmethode die Auflösungsantwort an

die Quell-Spoke.



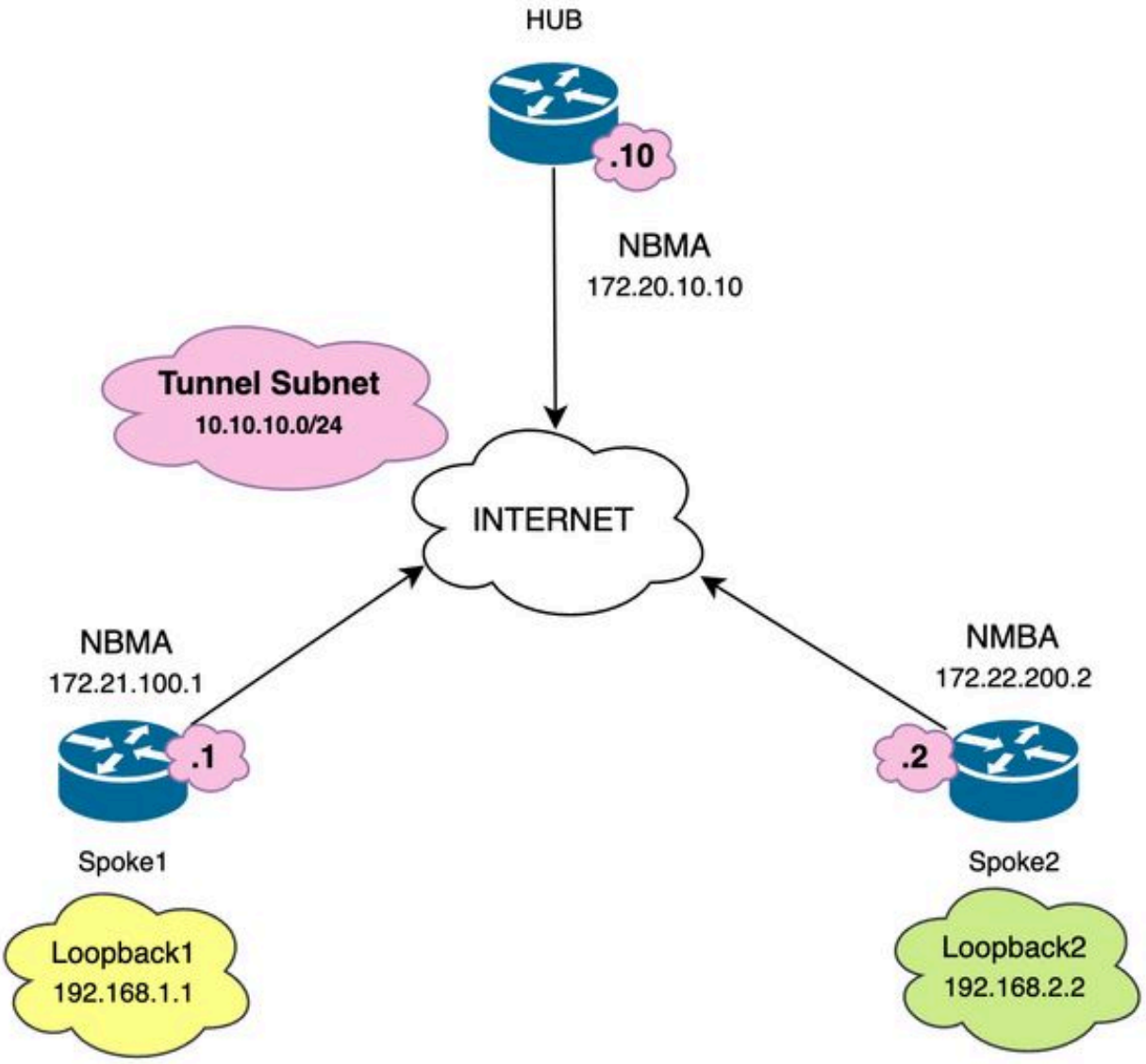
NHRP-Nachrichtenfluss zwischen den Spokes in Phase 2



Hinweis: Bevor der Auflösungsprozess beginnen kann, müssen alle Speichen bereits beim HUB registriert sein.

Topologie

Dieses Diagramm zeigt die Topologie, die für dieses Szenario verwendet wird:



Netzwerkdiagramm und verwendete IP-Subnetze

Schritte zur Fehlerbehebung

In diesem Szenario ist der Spoke-to-Spoke-Tunnel zwischen Spoke1 und Spoke2 nicht eingerichtet. Dies wirkt sich auf die Kommunikation zwischen den lokalen Ressourcen aus (dargestellt durch Loopback-Schnittstellen), da diese sich nicht erreichen können.

```
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)
```

Erste Validierung

Bei einem solchen Szenario ist es wichtig, zunächst die Tunnelkonfiguration zu validieren und sicherzustellen, dass beide Geräte die richtigen Werte enthalten. Um die Tunnelkonfiguration zu überprüfen, führen Sie den Befehl show running-config interface tunnel<ID> aus.

Tunnelkonfiguration für 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
```

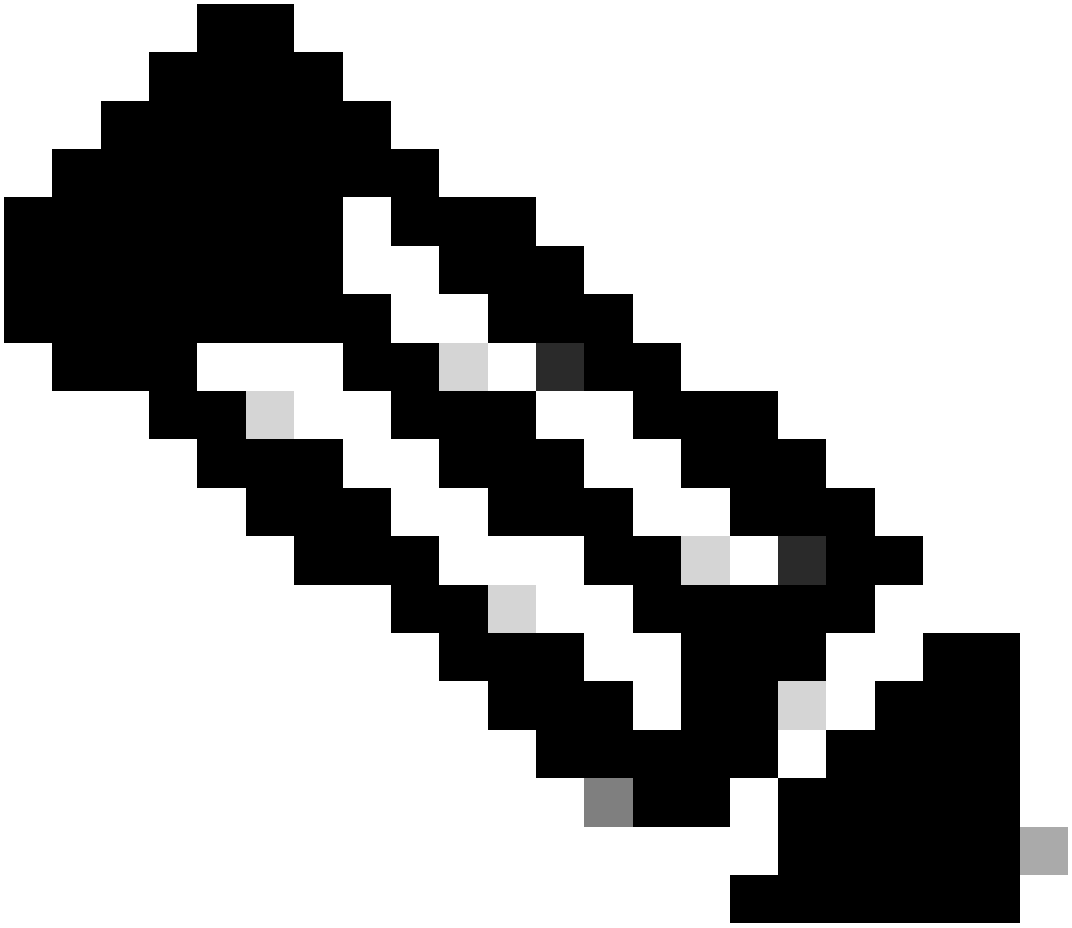
Konfiguration des Spoke 2-Tunnels:

<#root>

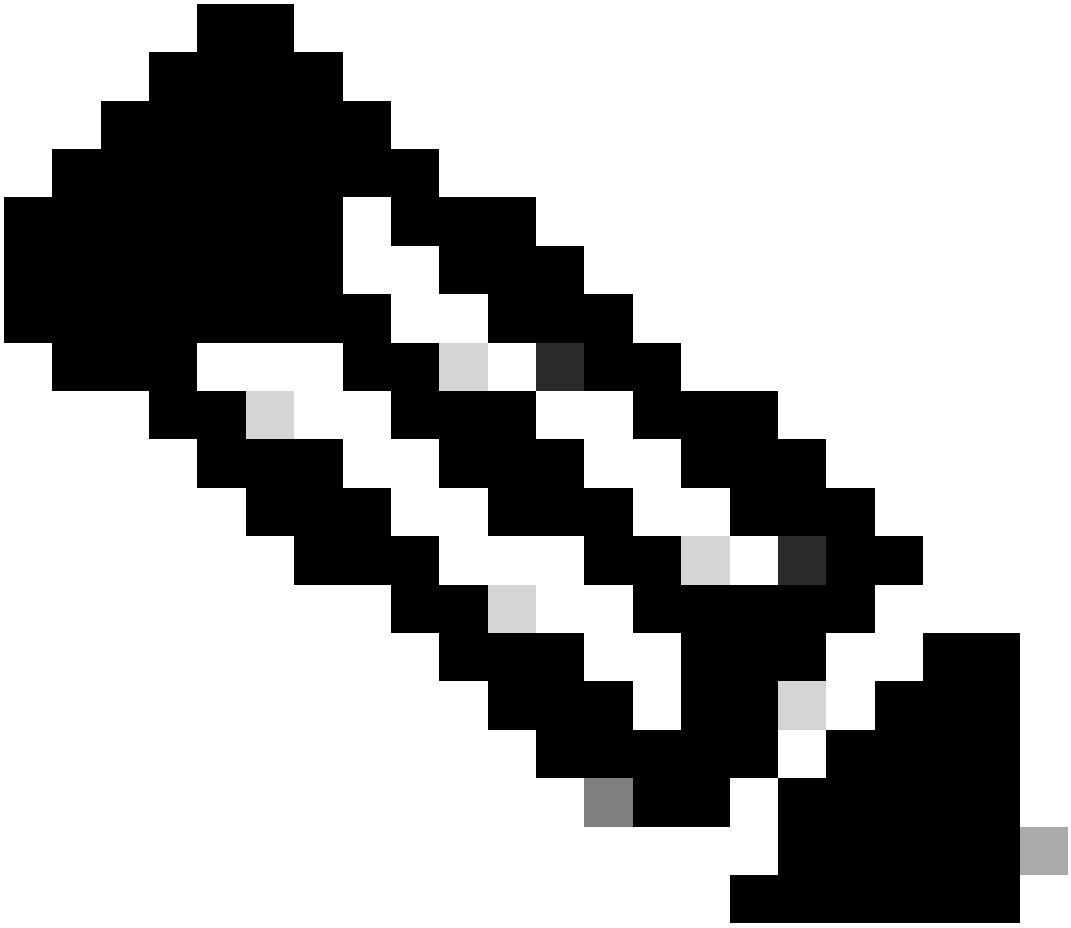
```
SPOKE2#show running-config interface tunnel10
Building configuration...

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
```

Bei der Konfiguration müssen Sie überprüfen, ob die Zuordnung zum HUB korrekt ist, ob die NHRP-Authentifizierungszeichenfolge zwischen den Geräten übereinstimmt, ob für beide Stationen dieselbe DMVPN-Phase konfiguriert ist und ob bei Verwendung des IPSEC-Schutzes die richtige Kryptografiekonfiguration angewendet wurde.

Wenn die Konfiguration korrekt ist und IPSEC-Schutz umfasst, muss sichergestellt werden, dass die IKE- und IPSEC-Protokolle ordnungsgemäß funktionieren. Dies liegt daran, dass NHRP den IPSEC-Tunnel als Transportmethode für die vollständige Aushandlung verwendet. Um den Status der IKE/IPSEC-Protokolle zu überprüfen, führen Sie den Befehl show crypto IPSEC as peer x.x.x.x aus (wobei x.x.x.x die NBMA-IP-Adresse der Spoke ist, mit der Sie den Tunnel einzurichten versuchen).



Hinweis: Um zu überprüfen, ob der IPSEC-Tunnel in Betrieb ist, müssen die Tunnelinformationen im Abschnitt für die ein- und ausgehende Encapsulation Security Payload (ESP) vorhanden sein (SPI, Transformationssatz usw.). Alle in diesem Abschnitt gezeigten Werte müssen an beiden Enden übereinstimmen.



Hinweis: Wenn Probleme mit IKE/IPSEC identifiziert werden, muss sich die Fehlerbehebung auf diese Protokolle konzentrieren.

IKE/IPSEC-Tunnelstatus auf Spoke 1:

<#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:

IKE/IPSEC-Tunnelstatus auf Spoke 2:

<#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:
```

Die Ausgabe zeigt an, dass der IPSEC-Tunnel auf beiden Stationen aktiv ist, Spoke2 jedoch verschlüsselte Pakete (Encaps), aber keine entschlüsselten Pakete (Decaps) anzeigt. In Spoke1 werden dagegen keine Pakete angezeigt, die durch den IPSEC-Tunnel fließen. Dies weist darauf hin, dass das Problem beim NHRP-Protokoll auftreten kann.

Tools für die Fehlerbehebung

Nachdem Sie die anfängliche Validierung durchgeführt und die Konfiguration bestätigt haben und die IKE/IPSEC-Protokolle (falls erforderlich) das Kommunikationsproblem nicht verursachen, können Sie die in diesem Abschnitt beschriebenen Tools verwenden, um die Fehlerbehebung fortzusetzen.

Nützliche Befehle

Mit dem Befehl show dmvpn interface tunnel<ID> erhalten Sie DMVPN-spezifische Sitzungsinformationen (NBMA-/Tunnel-IP-Adressen, Tunnelstatus, Up-/Down-Zeit und Attribut). Sie können das detail-Schlüsselwort verwenden, um Details aus der Crypto-Sitzung bzw. dem Socket anzuzeigen. Es ist wichtig zu erwähnen, dass der Zustand des Tunnels an beiden Enden übereinstimmen muss.

Spoke 1 zeigt dmvpn interface tunnel<ID>-Ausgabe an:

<#root>

```
SPOKE1#

show dmvpn interface tunnel10


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

```
UpDn Time --> Up or Down Time for a Tunnel
========================================================================

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

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

172.20.10.10 10.10.10.2 UP 00:00:51 I2

```
                        10.10.10.10     UP  02:53:27  S
```

Spoke 2 zeigt dmvpn interface tunnel<ID>-Ausgabe an:

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

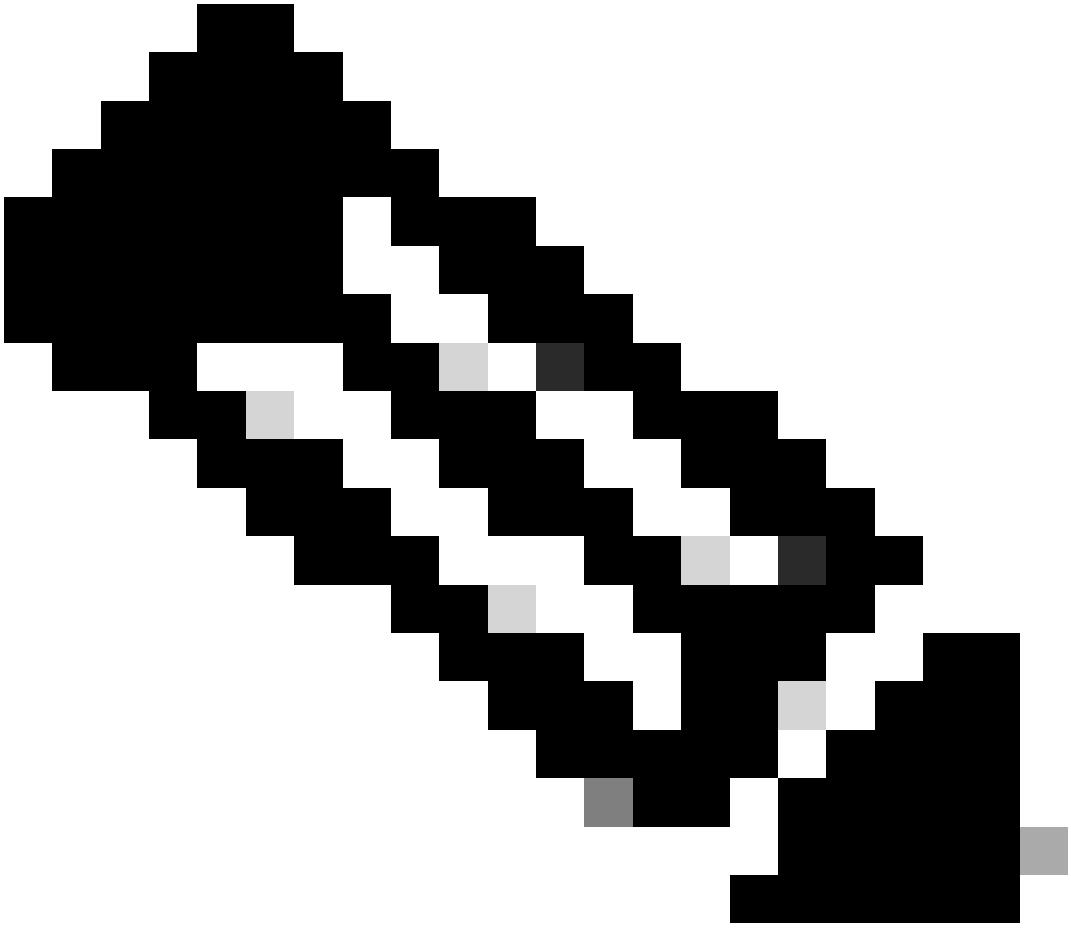
1 172.21.100.1 10.10.10.1 UP 00:03:53 D

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

Die Ausgabe auf jedem Gerät zeigt unterschiedliche Informationen für jede Speiche an. In der Tabelle Spoke1 sehen Sie, dass der Eintrag für Spoke 2 nicht die richtige NBMA-IP-Adresse enthält und das Attribut unvollständig (I2) erscheint. Andererseits zeigt die Tabelle Spoke2 die korrekte Zuordnung (NBMA/Tunnel-IP-Adressen) und den Status up an, was darauf hinweist, dass der Tunnel vollständig ausgehandelt ist.

Die folgenden Befehle können bei der Fehlerbehebung hilfreich sein:

- show ip nhrp: NHRP-Zuordnungsinformationen anzeigen
- show ip nhrp traffic interface tunnel10: Zeigt NHRP-Datenverkehrsstatistiken an

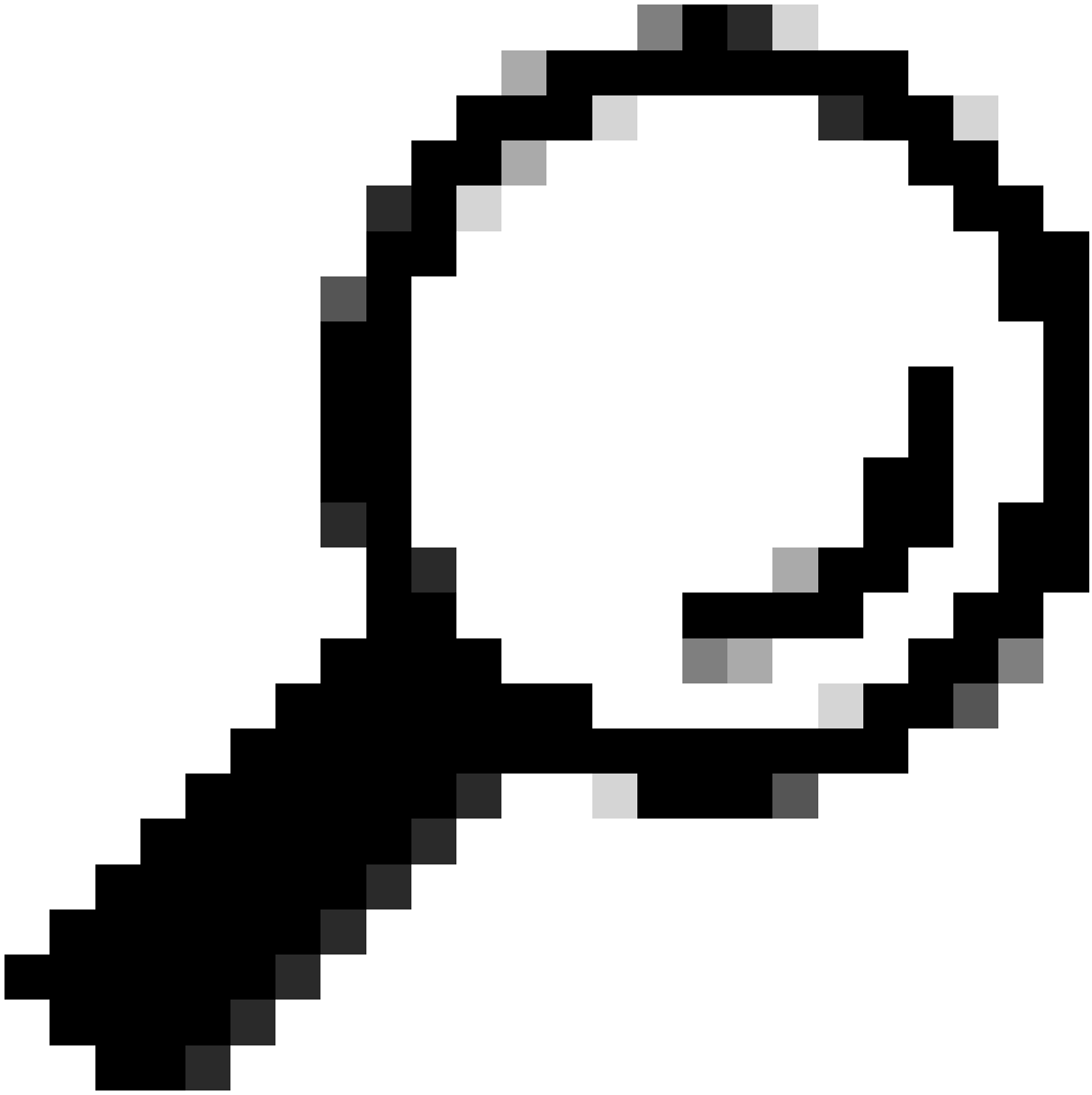


Hinweis: Spezifikationen für Befehle (Syntax, Beschreibung, Schlüsselwörter, Beispiel) finden Sie in der Befehlsreferenz: Cisco IOS Security Command Reference: Commands S to Z (Cisco IOS-Sicherheitsreferenz)

Fehlerbehebung

Nachdem Sie die vorherigen Informationen überprüft und bestätigt haben, dass im Tunnel Verhandlungsprobleme auftreten, müssen Sie Debug-Meldungen aktivieren, um zu beobachten, wie die NHRP-Pakete ausgetauscht werden. Die nächsten Fehlerbehebungen müssen auf allen beteiligten Geräten aktiviert werden:

1. debug dmvpn condition peer NBMA x.x.x.x (wobei x.x.x.x die IP-Adresse des Remote-Geräts ist).
2. debug dmvpn all: Dieser Befehl aktiviert die ISAKMP-, IKEv2-, IPSEC-, DMVPN- und NHRP-Debugging-Befehle.



Tipp: Es wird empfohlen, den Befehl peer condition jedes Mal zu verwenden, wenn Sie die Debugs aktivieren, damit Sie die Aushandlung des jeweiligen Tunnels sehen können.

Um den vollständigen NHRP-Fluss anzuzeigen, wurden auf jedem Gerät die folgenden Debugging-Befehle verwendet:

Speiche 1

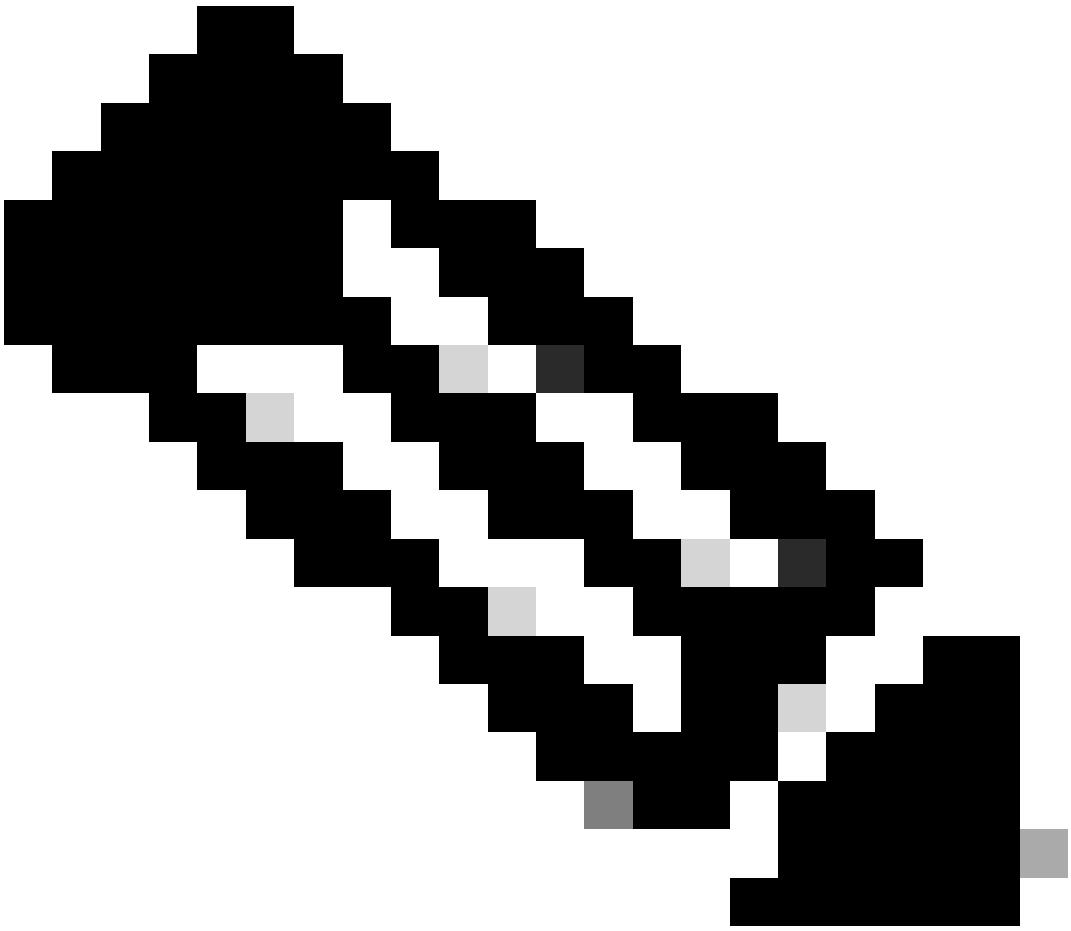
```
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```

HUB

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

Speiche 2

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



Hinweis: Die Debug-Meldungen müssen auf allen beteiligten Geräten gleichzeitig aktiviert

---

und erfasst werden.

Auf allen Geräten aktivierte Debugs werden mit dem Befehl show debug angezeigt:

<#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
```

Nachdem Sie alle Debug-Meldungen gesammelt haben, müssen Sie mit der Analyse der Debug-Meldungen in der Source-Spoke (Spoke1) beginnen. Dadurch können Sie die Aushandlung von Anfang an verfolgen.

Spoke1-Debug-Ausgabe:

<#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
```

Sobald der Spoke1 NHRP-Prozess beginnt, zeigen die Protokolle an, dass das Gerät die NHRP-Auflösungsanforderung sendet. Das Paket enthält einige wichtige Informationen wie den src NMBA und das src-Protokoll, die die NBMA-IP-Adresse und die Tunnel-IP-Adresse der Quell-Spoke (Spoke1) sind. Sie können auch den dst-Protokollwert sehen, der die Tunnel-IP-Adresse der Ziel-Spoke (Spoke2) enthält. Dies zeigt an, dass Spoke1 die NBMA-Adresse von Spoke2 abfragt, um die Zuordnung abzuschließen. Ebenfalls auf dem Paket können Sie den erforderlichen Wert finden, der Ihnen helfen kann, das Paket entlang des Pfads zu verfolgen. Dieser Wert bleibt während des gesamten Prozesses gleich und kann beim Verfolgen eines bestimmten Ablaufs der NHRP-Aushandlung hilfreich sein. Das Paket enthält weitere Werte, die für die Aushandlung wichtig sind, z. B. die NHRP-Authentifizierungszeichenfolge.

Nachdem das Gerät die NHRP-Auflösungsanforderung gesendet hat, zeigen die Protokolle an, dass eine erneute Übertragung gesendet wird. Dies liegt daran, dass das Gerät die NHRP-Auflösungsantwort nicht sieht und das Paket daher erneut sendet. Da Spoke1 die Antwort nicht sieht, muss das Paket auf dem nächsten Gerät im Pfad, d. h. dem HUB, nachverfolgt werden.

HUB-Debug-Ausgabe:

<#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

Anhand des Werts der Anforderung können Sie feststellen, dass das HUB die von Spoke1 gesendete Auflösungsanforderung empfängt. Im Paket sind die Werte von src NBMA und src Protokoll die Informationen von Spoke1, und der Wert von dst Protokoll ist die Tunnel-IP von Spoke2, wie es auf den Fehlerbehebungen von Spoke1 zu sehen war. Wenn der HUB die Auflösungsanforderung empfängt, führt er eine Routensuche durch und leitet das Paket an Spoke2 weiter. Dem weitergeleiteten Paket fügt der HUB eine Erweiterung mit eigenen Informationen hinzu (NBMA-IP-Adresse und Tunnel-IP-Adresse).

Die vorherigen Fehlerbehebungen zeigen, dass der HUB die Auflösungsanforderung korrekt an Spoke 2 weiterleitet. Der nächste Schritt besteht daher darin, zu bestätigen, dass Spoke2 das Paket empfängt, es korrekt verarbeitet und die Lösungsantwort an Spoke1 sendet.

Spoke2-Debug-Ausgabe:

<#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
```

Die Anforderung entspricht dem Wert aus den vorherigen Ausgaben. Dadurch wird bestätigt, dass das von Spoke1 gesendete NHRP-Auflösungsanforderungspaket Spoke2 erreicht. Dieses Paket löst eine Routensuche für Spoke2 aus und stellt fest, dass die Auflösungsanforderung für sich selbst gilt. Spoke2 fügt daher die Informationen von Spoke1 seiner NHRP-Tabelle hinzu. Bevor das Auflösungsantwortpaket an Spoke1 zurückgesendet wird, fügt das Gerät seine eigenen Informationen (NBMA-IP-Adresse und Tunnel-IP-Adresse) hinzu, sodass Spoke1 dieses Paket verwenden kann, um diese Informationen seiner Datenbank hinzuzufügen.

Basierend auf allen gefundenen Fehlerbehebungen erreicht die von Spoke2 gesendete Antwort

auf die NHRP-Auflösung nicht Spoke1. Der HUB kann verworfen werden, da er das NHRP Resolution Request-Paket empfängt und wie erwartet weiterleitet. Der nächste Schritt ist daher, Aufnahmen zwischen Spoke1 und Spoke2 zu machen, um weitere Details zu dem Problem zu erhalten.

Integrierte Paketerfassung

Die integrierte Paketerfassungsfunktion ermöglicht Ihnen die Analyse des Datenverkehrs, der durch das Gerät läuft. Der erste Schritt zur Konfiguration besteht darin, eine Zugriffsliste zu erstellen, die den Datenverkehr enthält, den Sie bei beiden Datenflüssen (ein- und ausgehend) erfassen möchten.

In diesem Szenario werden die NBMA-IP-Adressen verwendet:

```
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
```

Konfigurieren Sie dann die Erfassung mit dem Befehl monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 interface <WAN_INTERFACE>, und starten Sie die Erfassung mit dem Befehl monitor capture <CAPTURE_NAME> start.

Erfassungskonfiguration für Spoke1 und Spoke2:

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

Um die Ausgabe der Erfassung anzuzeigen, verwenden Sie den Befehl show monitor capture <CAPTURE_NAME> buffer brief.

Erfassungsausgabe 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
```

Erfassungsausgabe Spoke2:

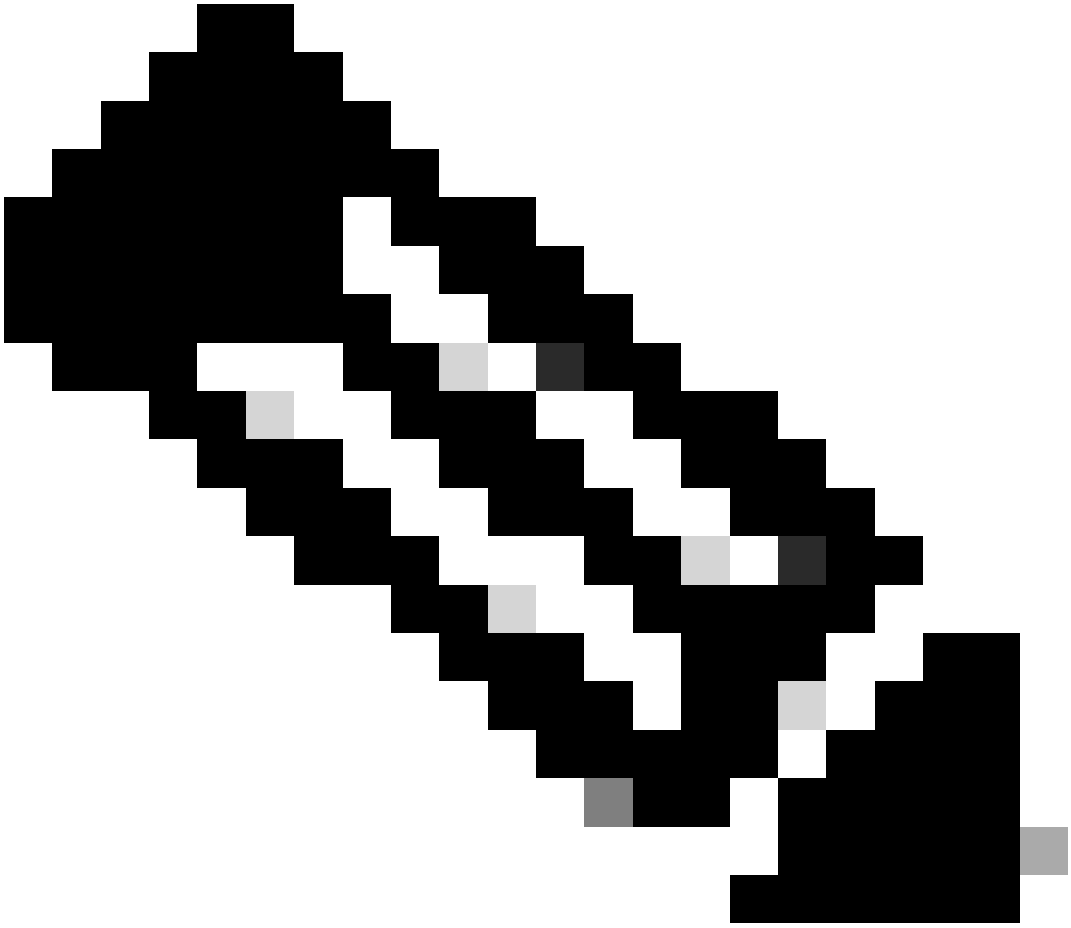
<#root>

```
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
```

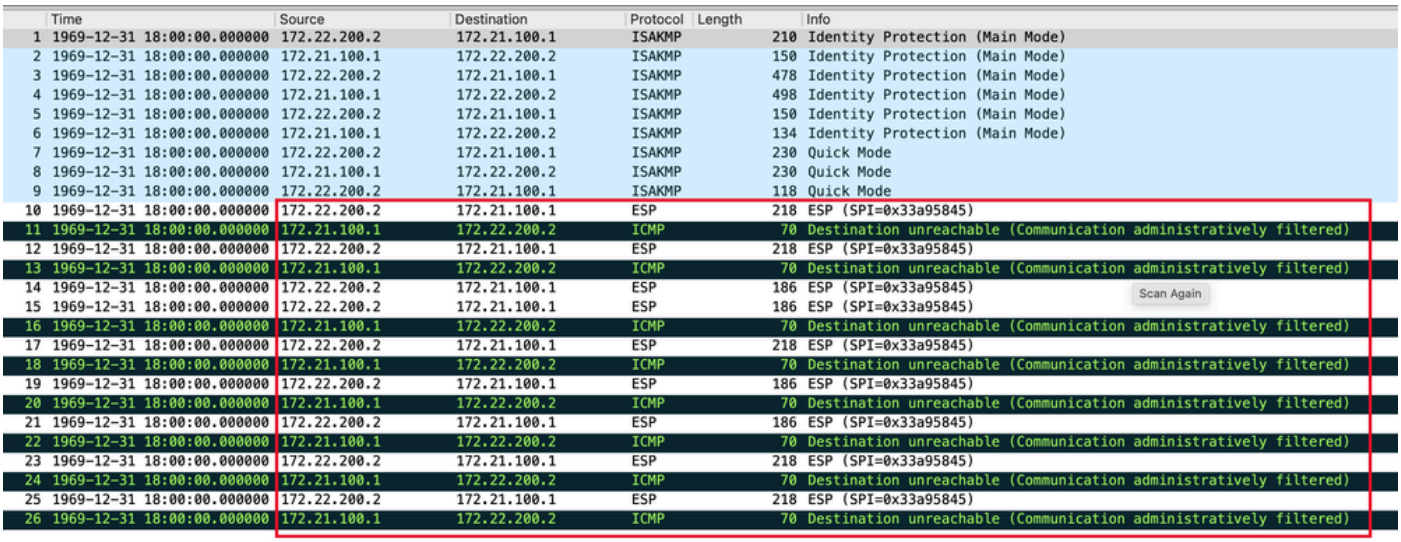
Die Ausgabe der Captures zeigt an, dass die ursprünglichen Pakete UDP-Datenverkehr sind und zeigt damit die IKE/IPSEC-Aushandlung an. Danach sendet Spoke2 die Auflösungsantwort an Spoke1, die als ESP-Datenverkehr (Paket 9) angesehen wird. Danach lautet der erwartete Datenverkehrsfluss ESP. Als Nächstes wird jedoch ICMP-Datenverkehr von Spoke1 nach Spoke2 erkannt.

Um das Paket genauer zu analysieren, können Sie die pcap-Datei vom Gerät exportieren, indem Sie den Befehl show monitor capture <CAPTURE_NAME> buffer dump ausführen. Dann verwenden Sie ein Decoder-Tool, um die Dump-Ausgabe in eine pcap-Datei zu konvertieren, sodass Sie es mit Wireshark öffnen können.



Hinweis: Cisco bietet einen Paket-Analyzer, wo Sie die Erfassungskonfiguration, Beispiele und einen Decoder finden können: [Cisco TAC Tool - Packet Capture Config Generator and Analyzer]

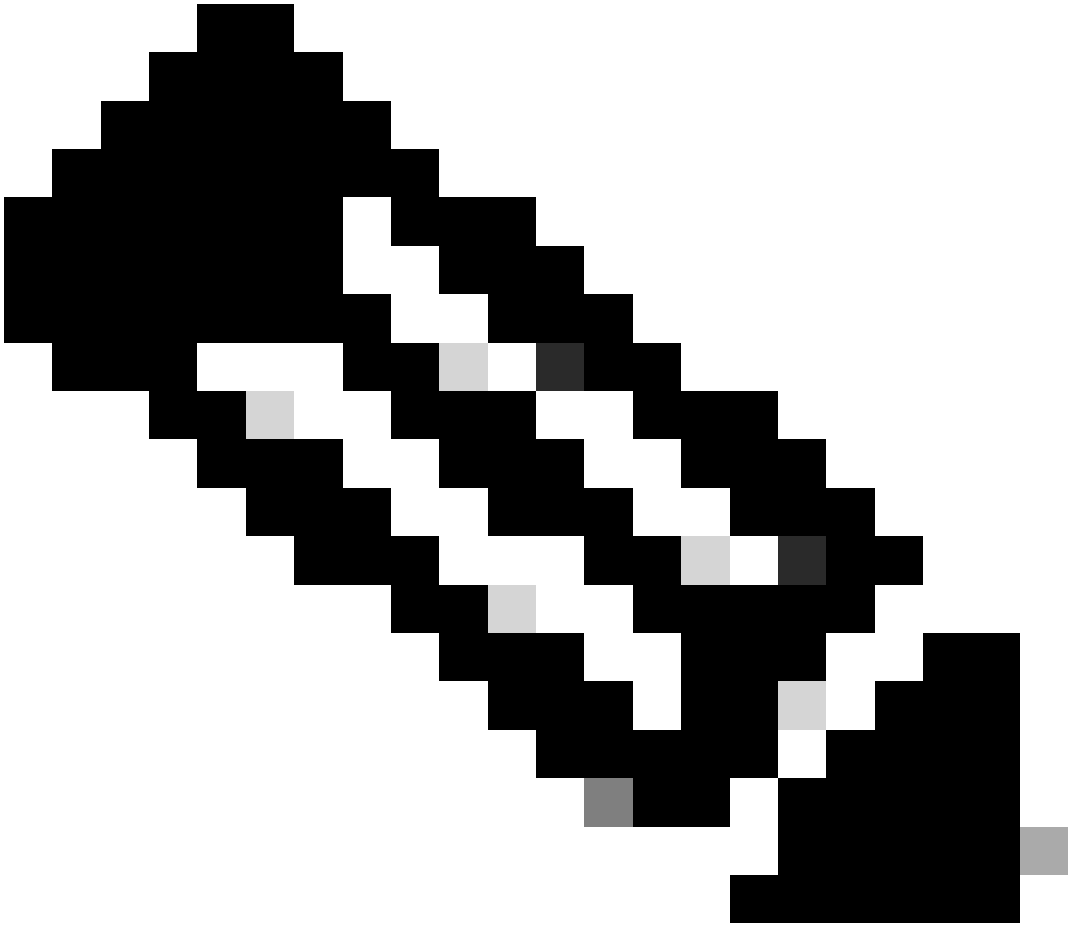
Wireshark-Ausgabe:



	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)

Ausgabe in Wireshark erfassen

Der Inhalt des ICMP-Pakets enthält die Fehlermeldung Destination unreachable (Kommunikation administrativ gefiltert). Dies weist darauf hin, dass eine Art Filter vorhanden ist, z. B. eine Router-ACL oder eine Firewall, die sich auf den Datenverkehr entlang des Pfads auswirkt. In den meisten Fällen ist der Filter auf dem Gerät konfiguriert, das das Paket sendet (in diesem Fall Spoke1), aber auch mittlere Geräte können es senden.



Hinweis: Die Ausgabe von Wireshark ist auf beiden Stationen gleich.

Cisco IOS® XE DataPath Packet Trace-Funktion

Die Datenpfad-Paketablaufverfolgungsfunktion von Cisco IOS XE analysiert, wie das Gerät den Datenverkehr verarbeitet. Um sie zu konfigurieren, müssen Sie eine Zugriffsliste erstellen, die den Datenverkehr enthält, den Sie bei beiden Datenverkehrsflüssen (ein- und ausgehend) erfassen möchten.

In diesem Szenario werden die NBMA-IP-Adressen verwendet.

```
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
```

Konfigurieren Sie anschließend das fia-trace-Feature, und legen Sie die Debugbedingung so fest, dass die Zugriffsliste verwendet wird. Beginnen Sie schließlich mit der Bedingung.

```
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: Ermöglicht die detaillierte fia-trace und stoppt sie, sobald die Anzahl der konfigurierten Pakete erfasst wurde
- debug platform condition ipv4 access-list <ACL-NAME> both: Legt eine Bedingung für das Gerät mithilfe der zuvor konfigurierten Zugriffsliste fest.
- debug platform condition start: startet die Bedingung

Um die Ausgabe von fia-trace zu überprüfen, verwenden Sie die nächsten Befehle.

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

Spoke1 zeigt eine Plattform-Paketverfolgungsstatistik an:

<#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
```

In der Ausgabe von show platform packet-trace statistics werden die Zähler für die vom Gerät verarbeiteten Pakete angezeigt. Auf diese Weise können Sie die ein- und ausgehenden Pakete anzeigen und überprüfen, ob das Gerät Pakete verwirft, und den Grund dafür angeben.

In der gezeigten Ausgabe verwirft Spoke1 einige Pakete mit der Beschreibung Ipv4Acl. Um diese Pakete weiter zu analysieren, kann der Befehl show platform packet-trace summary verwendet werden.

Spoke1 zeigt eine Plattformpaket-Ablaufverfolgungs-Zusammenfassung an:

<#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
```

Mit dieser Ausgabe können Sie sehen, wie jedes Paket das Gerät erreicht und verlässt, sowie Eingangs- und Ausgangsschnittstellen. Außerdem wird der Status des Pakets angezeigt. Dieser gibt an, ob es weitergeleitet, verworfen oder intern verarbeitet wurde (Punkt).

In diesem Beispiel hat diese Ausgabe dazu beigetragen, die Pakete zu identifizieren, die vom Gerät verworfen werden. Mit dem Befehl show platform packet-trace packet <PACKET_NUMBER> können Sie sehen, wie das Gerät dieses Paket verarbeitet.

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

<#root>

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

Output : GigabitEthernet1

State : DROP 8 (Ipv4Acl)

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

Input : GigabitEthernet1

     Output : <unknown>
```

Source : 172.22.200.2

Destination : 172.21.100.1

Protocol : 50 (ESP)

Feature: DEBUG_COND_INPUT_PKT
Entry : Input - 0x812707d0

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

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

Input : GigabitEthernet1

Output : <unknown>

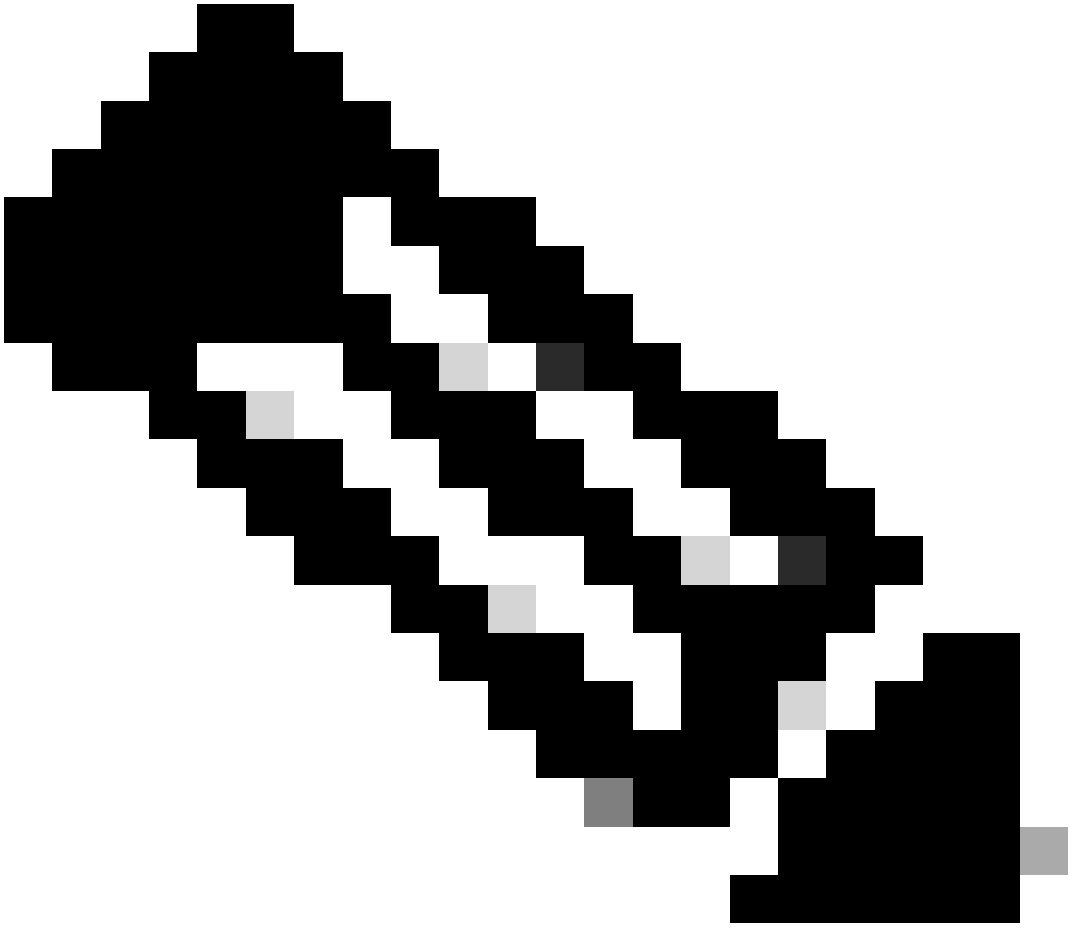
Lapsed time : 174 ns
Feature: INPUT_DROP

```
     Entry : Input - 0x8126e568

Input : GigabitEthernet1

     Output : <unknown>

     Lapsed time : 116 ns

Feature: IPV4_INPUT_ACL

     Entry : Input - 0x81271f70

Input : GigabitEthernet1

     Output : <unknown>

     Lapsed time : 12915 ns
```

Im ersten Teil sehen Sie die Eingangs- und Ausgangsschnittstelle und den Status des Pakets. Darauf folgt der zweite Teil der Ausgabe, in dem Sie die Quell- und Ziel-IP-Adresse sowie das Protokoll finden.

Jede nachfolgende Phase zeigt, wie das Gerät dieses Paket verarbeitet. Dies bietet Einblicke in Konfigurationen wie Network Address Translation (NAT), Zugriffslisten oder andere Faktoren, die sich auf diese auswirken können.

In diesem Fall kann erkannt werden, dass das Protokoll des Pakets ESP ist, die Quell-IP die NBMA-IP-Adresse von Spoke2 ist und die Ziel-IP die NBMA-IP-Adresse von Spoke1 ist. Dies weist darauf hin, dass dieses Paket in der NHRP-Aushandlung fehlt. Außerdem wird beobachtet, dass in keiner Phase eine Ausgangsschnittstelle angegeben wird, was darauf hindeutet, dass sich etwas auf den Datenverkehr ausgewirkt hat, bevor er weitergeleitet werden konnte. In der vorletzten Phase können Sie sehen, dass das Gerät den eingehenden Datenverkehr auf der angegebenen Schnittstelle (GigabitEthernet1) verwirft. In der letzten Phase wird eine Zugriffsliste für Eingaben angezeigt, die andeutet, dass die Schnittstelle konfiguriert werden kann, um den Ausfall zu verursachen.



Hinweis: Wenn die in der Verhandlung involvierten Stationen nach der Verwendung aller in diesem Dokument aufgeführten Tools zur Fehlerbehebung keine Anzeichen dafür zeigen, dass sie den Datenverkehr verwerfen oder beeinflussen, ist die Fehlerbehebung auf diesen Geräten abgeschlossen.

Im nächsten Schritt müssen die dazwischen befindlichen mittleren Geräte (Firewalls, Switches, ISP) überprüft werden.

Lösung

Wenn ein solches Szenario eintritt, besteht der nächste Schritt darin, die in den vorherigen Ausgaben dargestellte Schnittstelle zu überprüfen. Dazu muss die Konfiguration überprüft werden, um festzustellen, ob der Datenverkehr beeinträchtigt wird.

Konfiguration der WAN-Schnittstelle:

<#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
```

Bei der Konfiguration der Schnittstelle wird eine Zugriffsgruppe angewendet. Es muss unbedingt sichergestellt werden, dass die in der Zugriffsliste konfigurierten Hosts den für die NHRP-Aushandlung verwendeten Datenverkehr nicht beeinträchtigen.

<#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)
```

Die zweite Aussage der Zugriffsliste besagt, dass die Kommunikation zwischen der NBMA-IP-Adresse von Spoke2 und der NBMA-IP-Adresse von Spoke1 abgelehnt wird, wodurch der zuvor erkannte Ausfall verursacht wird. Nachdem die Zugriffsgruppe von der Schnittstelle entfernt wurde, ist die Kommunikation zwischen den beiden Stationen erfolgreich:

```
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
```

Der IPSEC-Tunnel ist nun aktiv und zeigt Encaps und Decaps auf beiden Geräten an:

Speiche 1:

<#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:
```

Speiche 2:

<#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:
```

In der DMVPN-Tabelle von Spoke1 wird jetzt die richtige Zuordnung für beide Einträge angezeigt:

<#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
```

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.