Risoluzione dei problemi del tunnel Spoke-to-Spoke DMVPN fase 2

Sommario

Introduzione

In questo documento viene descritto come risolvere i problemi di un tunnel VPN spoke di fase 2 quando non viene stabilito.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei prossimi argomenti:

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

Componenti usati

Questo documento si basa sulla seguente versione del software:

- Cisco CSR1000V (VXE) - Versione 17.03.08

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico

ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

In questo documento viene descritto come configurare e utilizzare diversi strumenti di risoluzione dei problemi per un problema comune di DMVPN. Il problema è la negoziazione non riuscita di un tunnel DMVPN di fase 2, in cui l'origine ha parlato, lo stato DMVPN viene visualizzato con il mapping NBMA (Non-Broadcast Multi-Access)/Tunnel corretto alla destinazione spoke. Tuttavia, sul spoke di destinazione viene visualizzato un mapping errato.

Contesto teorico

È importante comprendere come vengono stabiliti i tunnel spoke-to-spoke quando si dispone di una configurazione DMVPN fase 2. Questa sezione fornisce una breve sintesi teorica del processo NHRP durante questa fase.

In DMVPN fase 2 è possibile creare tunnel spoke dinamici su richiesta. Ciò è possibile perché, su tutti i dispositivi all'interno del cloud DMVPN (hub e spoke) la modalità dell'interfaccia del tunnel cambia in multipoint GRE (Generic Routing Encapsulation). Una delle caratteristiche chiave di questa fase è che l'hub non viene percepito come l'hop successivo dagli altri dispositivi. Al contrario, tutti i raggi hanno reciprocamente le informazioni di routing. Quando si stabilisce un tunnel spoke-to-spoke nella fase 2, viene attivato un processo NHRP in cui gli spoke apprendono le informazioni sugli altri spoke e creano un mapping tra l'NBMA e gli indirizzi IP del tunnel.

Nelle fasi successive viene descritto come attivare il processo di risoluzione NHRP:

1. Quando l'origine spoke tenta di raggiungere la LAN della destinazione spoke, esegue una ricerca di route che attiva il messaggio di richiesta di risoluzione per ottenere l'indirizzo NBMA della destinazione spoke. L'origine spoke invia questo messaggio iniziale all'hub.

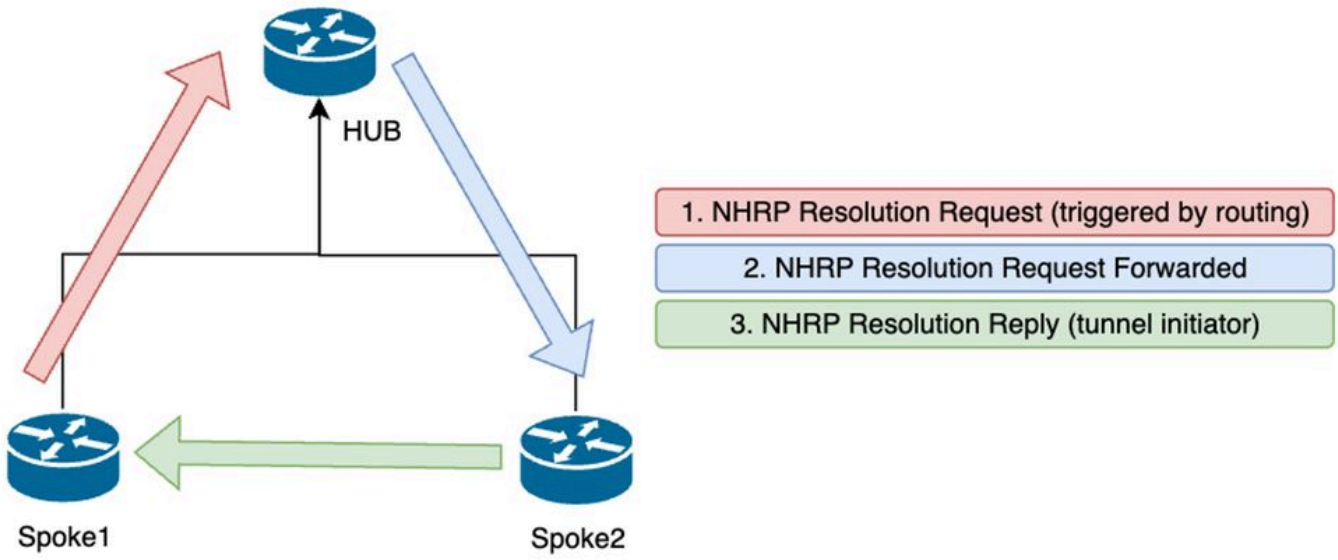
2. L'hub riceve la richiesta di risoluzione e la inoltra al spoke di destinazione.

3. Il spoke di destinazione invia la risposta di risoluzione al spoke di origine. Se alla configurazione del tunnel è collegato un profilo IPSEC:

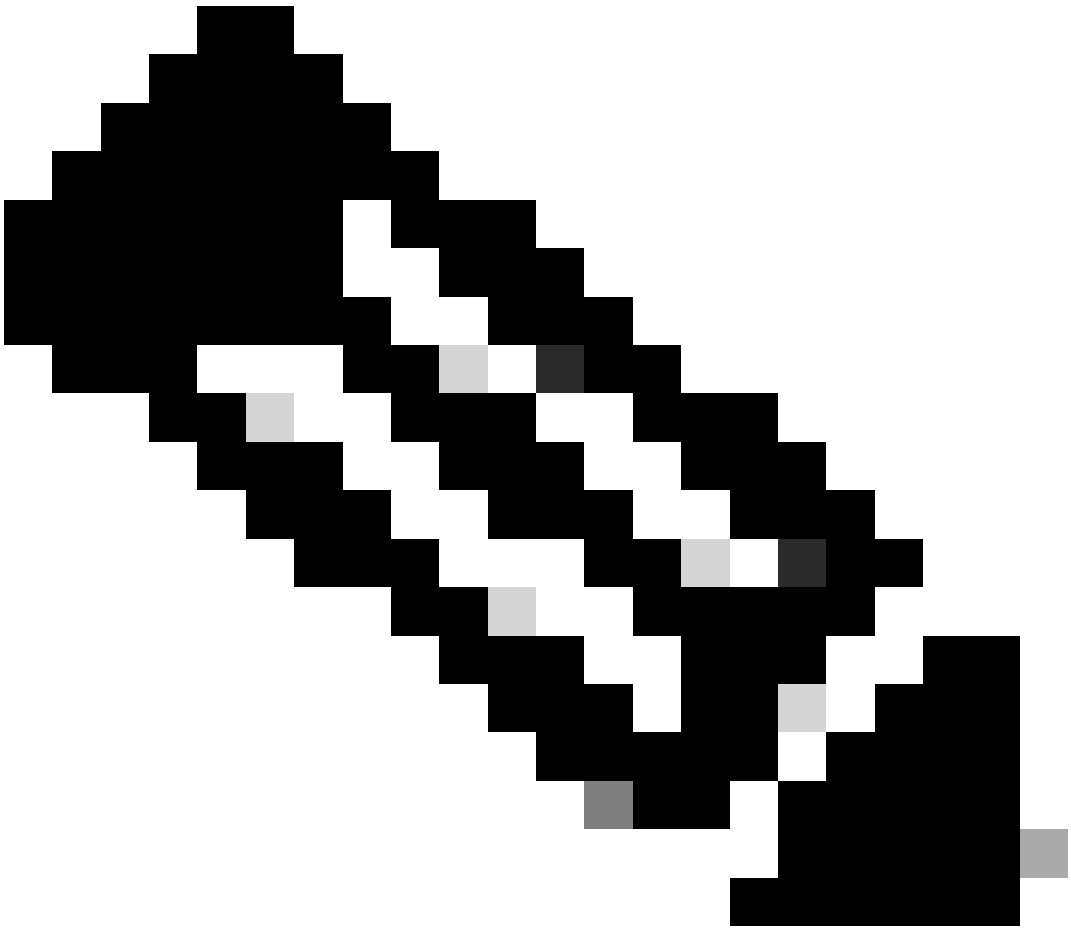
    - Il processo di risoluzione NHRP viene posticipato fino a quando non è possibile stabilire i protocolli IKE/IPSEC.

    - Il spoke di destinazione esegue l'inizializzazione e stabilisce i tunnel IKE/IPSEC.

    - Quindi, il processo NHRP viene ripreso e lo spoke di destinazione invia la risposta di risoluzione allo spoke di origine utilizzando il tunnel IPSEC come metodo di trasporto.



Flusso di messaggi NHRP tra i spoke nella fase 2



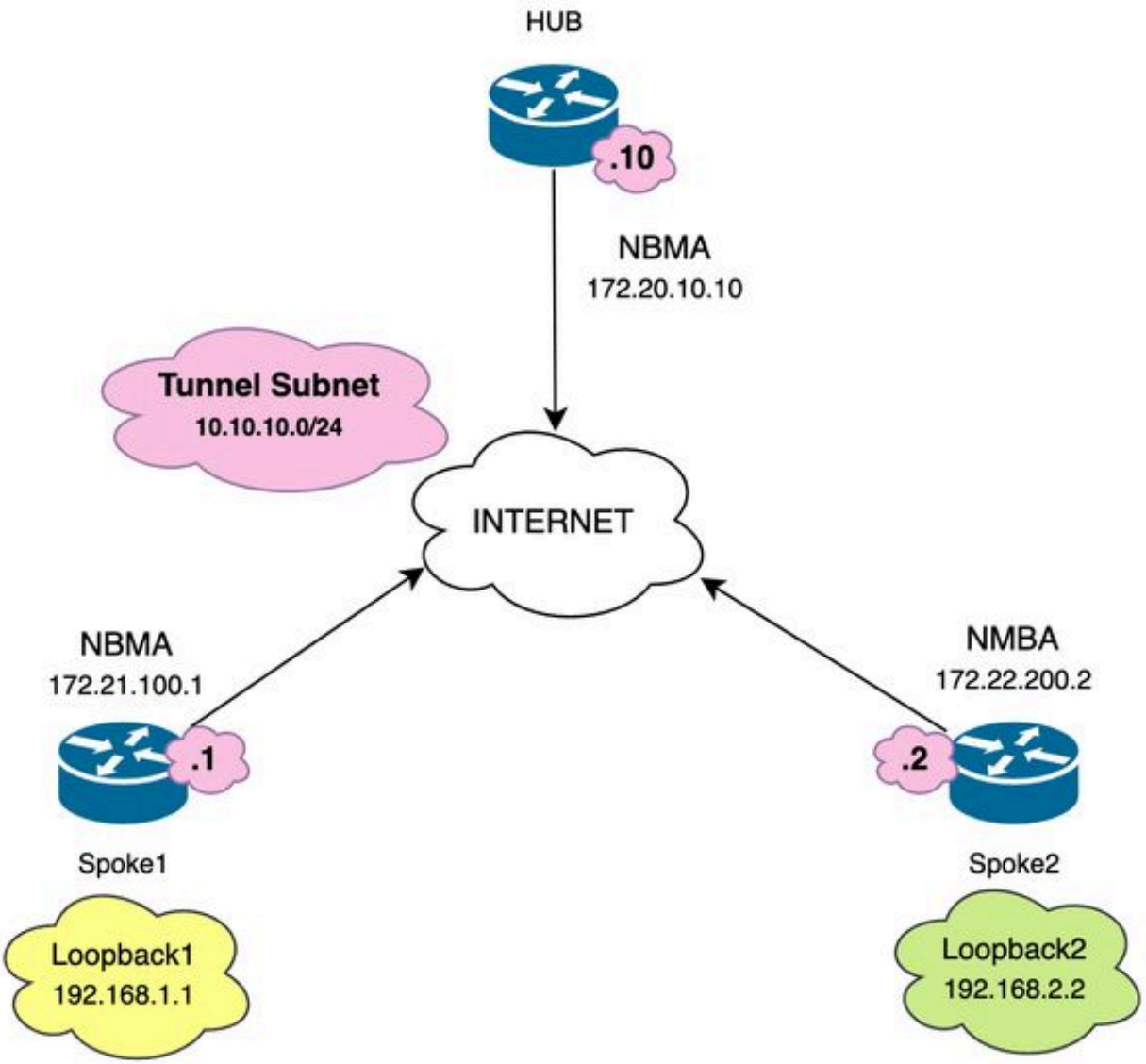
Nota: prima di poter avviare il processo di risoluzione, tutti i raggi devono essere già

---

registrati nell'HUB.

Topologia

Il diagramma mostra la topologia utilizzata per lo scenario:



Esempio di rete e subnet IP utilizzate

Procedura di risoluzione dei problemi

In questo scenario, il tunnel spoke tra Spoke1 e Spoke2 non viene stabilito, influendo sulla comunicazione tra le loro risorse locali (rappresentate da interfacce loopback) poiché non sono in grado di comunicare tra loro.

```
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)
```

Convalida iniziale

In questo caso, è importante iniziare a convalidare la configurazione del tunnel e verificare che entrambi i dispositivi includano i valori corretti. Per rivedere la configurazione del tunnel, eseguire il comando show running-config interface tunnel<ID>.

Configurazione tunnel Spoke 1:

<#root>

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

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


ip nhrp map 10.10.10.10 172.20.10.10


ip nhrp map multicast 172.20.10.10


ip nhrp network-id 10
ip nhrp nhs 10.10.10.10


tunnel source GigabitEthernet1
tunnel mode gre multipoint


tunnel protection IPSEC profile IPSEC_Profile_1


end
```

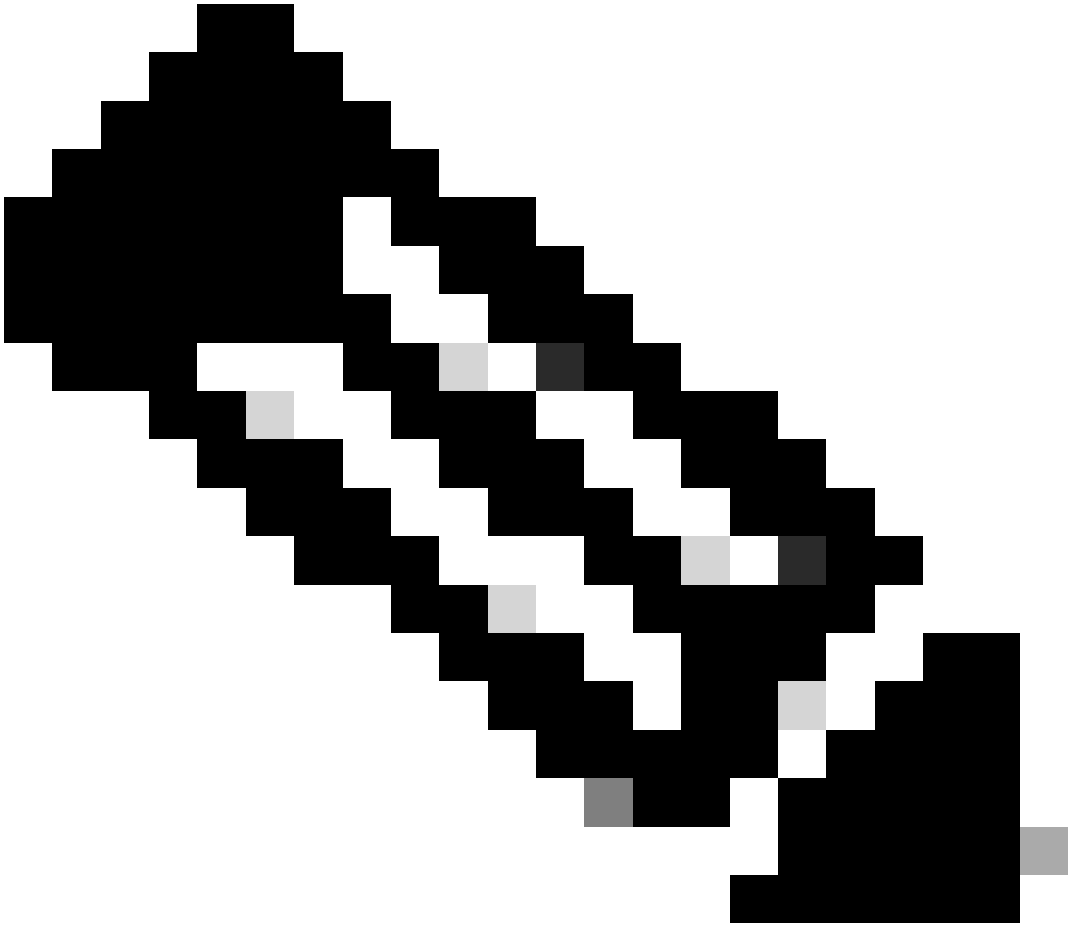
Configurazione tunnel Spoke 2:

<#root>

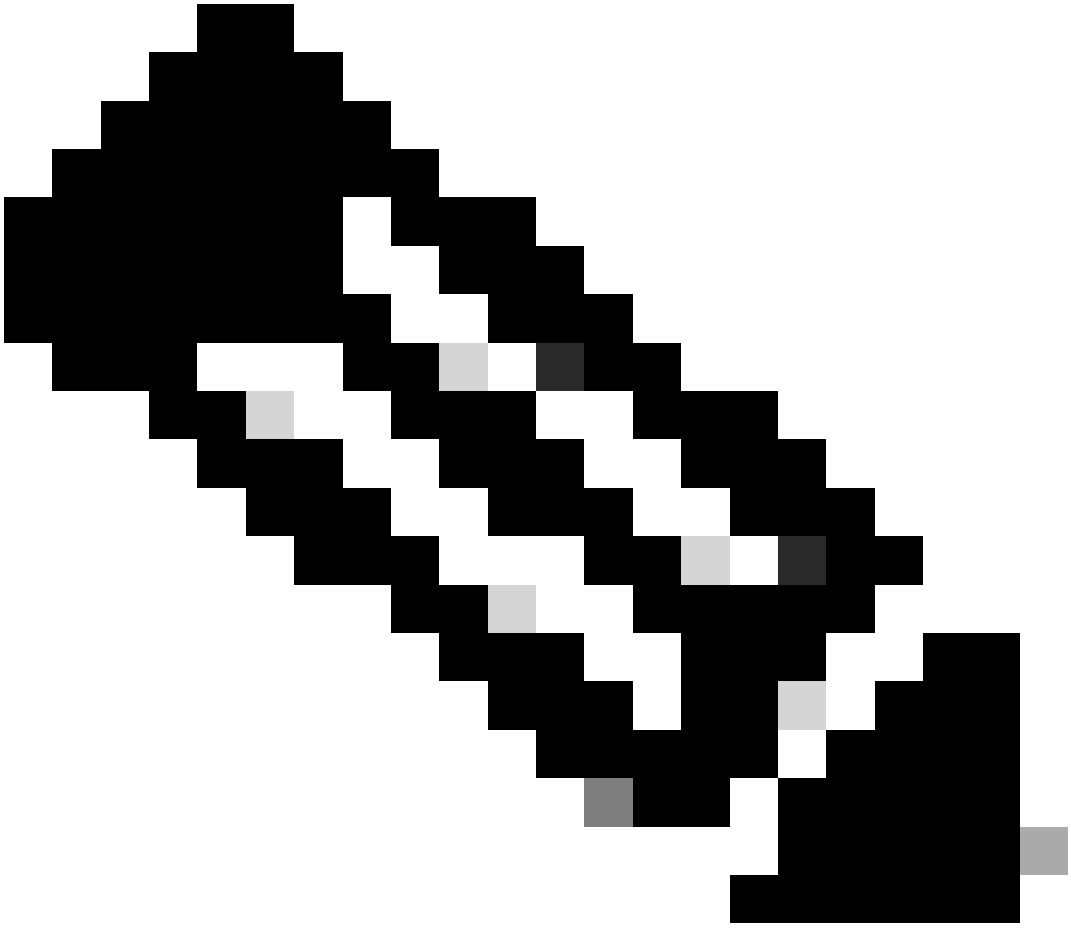
```
SPOKE2#show running-config interface tunnel10
Building configuration...

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


ip nhrp map 10.10.10.10 172.20.10.10


ip nhrp map multicast 172.20.10.10


ip nhrp network-id 10
ip nhrp nhs 10.10.10.10


tunnel source GigabitEthernet1
tunnel mode gre multipoint


tunnel protection IPSEC profile IPSEC_Profile_1


end
```

Sulla configurazione è necessario verificare che il mapping all'HUB sia corretto, che la stringa di autenticazione NHRP corrisponda tra i dispositivi, che entrambi gli spoke abbiano la stessa fase DMVPN configurata e, se viene utilizzata la protezione IPSEC, verificare che sia applicata la configurazione crittografica corretta.

Se la configurazione è corretta e include la protezione IPSEC, è necessario verificare che i protocolli IKE e IPSEC funzionino correttamente. Infatti, per la negoziazione completa, NHRP utilizza il tunnel IPSEC come metodo di trasporto. Per verificare lo stato dei protocolli IKE/IPSEC, eseguire il comando show crypto IPSEC sa peer x.x.x.x (dove x.x.x è l'indirizzo IP NBMA del spoke con cui si sta tentando di stabilire il tunnel).



Nota: per verificare se il tunnel IPSEC è attivo, la sezione ESP (Encapsulation Security Payload) in entrata e in uscita deve contenere le informazioni del tunnel (SPI, transform-set e così via). Tutti i valori mostrati in questa sezione devono corrispondere su entrambe le estremità.



Nota: se vengono identificati problemi con IKE/IPSEC, la risoluzione dei problemi deve essere incentrata su tali protocolli.

Stato tunnel IKE/IPSEC su Spoke1:

<#root>

SPOKE1#

show crypto IPSEC sa peer 172.22.200.2

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

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

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

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

#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0

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

inbound esp sas:

spi: 0x84502A19(2219846169)

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

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

inbound ah sas:

inbound pcp sas:

outbound esp sas:

spi: 0x6F6BF94A(1869347146)

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

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

outbound ah sas:

outbound pcp sas:

Stato tunnel IKE/IPSEC su Spoke2:

<#root>

SPOKE2#

show crypto IPSEC sa peer 172.21.100.1

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

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

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

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

#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0

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

inbound esp sas:

spi: 0x6F6BF94A(1869347146)

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

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

inbound ah sas:

inbound pcp sas:

outbound esp sas:

```
spi: 0x84502A19(2219846169)


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

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

outbound ah sas:

outbound pcp sas:
```

Gli output mostrano che su entrambi gli spoke il tunnel IPSEC è attivo, ma, Spoke2 mostra pacchetti crittografati (encaps) ma non pacchetti decriptati (decaps). Nel frattempo, Spoke1 non visualizza i pacchetti che passano attraverso il tunnel IPSEC. Ciò indica che il problema può essere causato dal protocollo NHRP.

Strumenti di risoluzione dei problemi

Dopo aver eseguito la convalida iniziale e aver confermato la configurazione e i protocolli IKE/IPSEC (se necessario) non causano il problema di comunicazione, è possibile utilizzare gli strumenti illustrati in questa sezione per continuare la risoluzione del problema.

Comandi utili

Il comando show dmvpn interface tunnel<ID> restituisce informazioni specifiche sulla sessione DMVPN (indirizzi IP NBMA/tunnel, stato del tunnel, tempo di attività/inattività e attributo). È possibile usare la parola chiave detail per visualizzare i dettagli della sessione di crittografia o del socket. È importante ricordare che lo stato del tunnel deve corrispondere su entrambi i lati.

Spoke 1 show dmvpn interface tunnel<ID> output:

<#root>

```
SPOKE1#

show dmvpn interface tunnel10


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

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

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

172.20.10.10 10.10.10.2 UP 00:00:51 I2

10.10.10.10 UP 02:53:27 S

Spoke 2 show dmvpn interface tunnel<ID> output:

<#root>

```
SPOKE2#
```

show dmvpn interface tunnel10

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

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

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

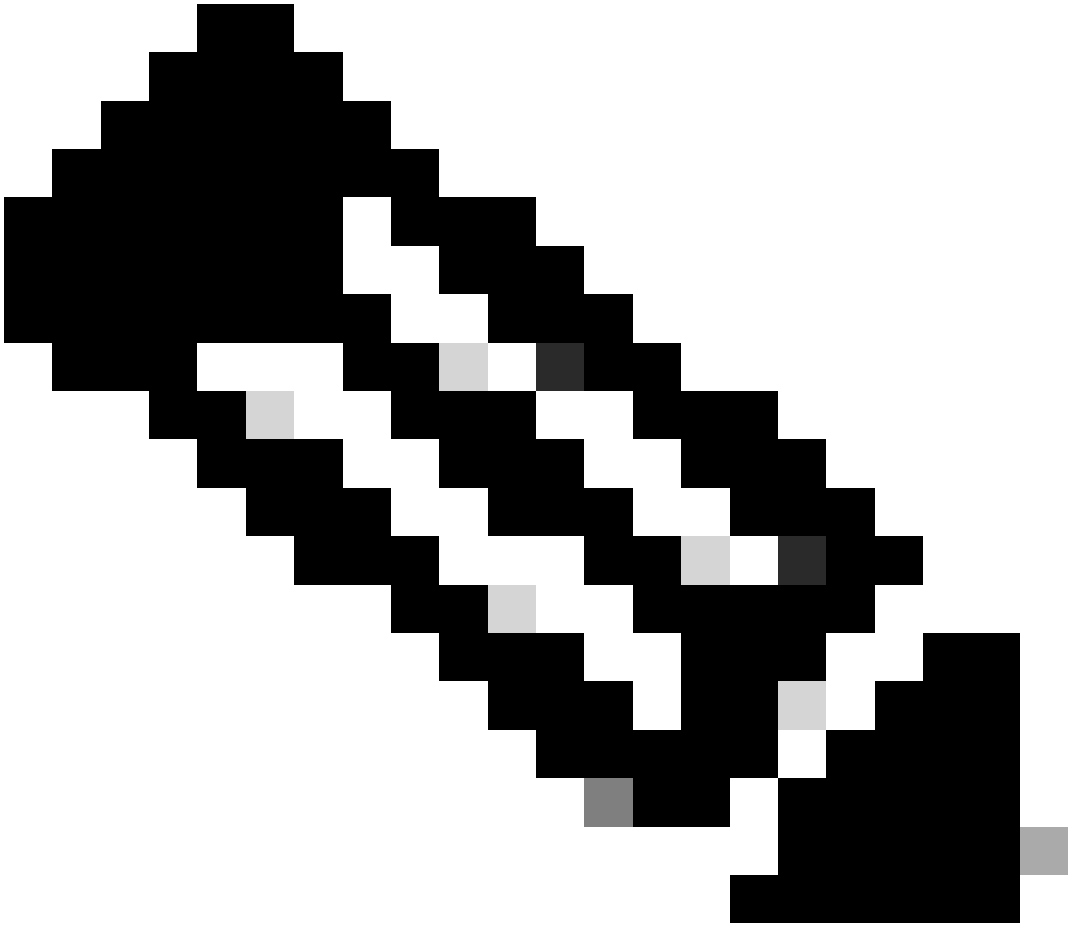
1 172.21.100.1 10.10.10.1 UP 00:03:53 D

1 172.20.10.10 10.10.10.10 UP 02:59:14 S

L'output su ciascun dispositivo mostra informazioni diverse per ciascun spoke. Nella tabella Spoke1, è possibile notare che la voce per Spoke 2 non include l'indirizzo IP NBMA corretto e che l'attributo è incompleto (I2). D'altra parte, la tabella Spoke2 mostra il mapping corretto (indirizzi IP NBMA/tunnel) e lo stato attivo che indica che il tunnel è stato completamente negoziato.

I comandi seguenti possono essere utili durante il processo di risoluzione dei problemi:

- show ip nhrp: visualizzazione delle informazioni di mapping NHRP
- show ip nhrp traffic interface tunnel10: visualizza le statistiche del traffico NHRP

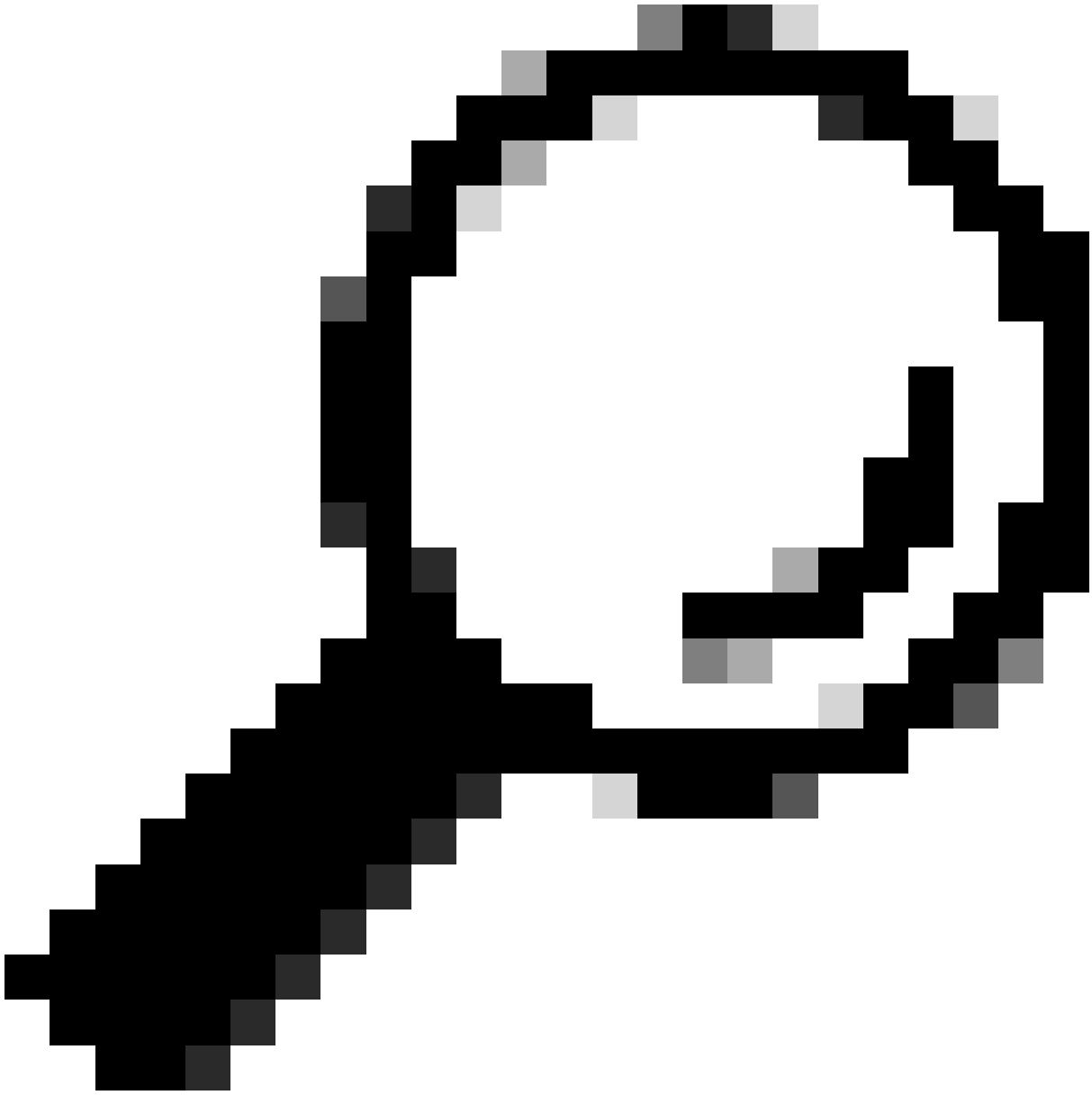


> Nota: per le specifiche dei comandi (sintassi, descrizione, parole chiave, esempio), consultare la Guida di riferimento dei comandi: Guida di riferimento dei comandi di Cisco IOS Security: comandi S-Z

Debug

Dopo aver verificato le informazioni precedenti e aver confermato che il tunnel sta incontrando problemi di negoziazione, è necessario abilitare i debug per osservare come vengono scambiati i pacchetti NHRP. I debug successivi devono essere abilitati su tutti i dispositivi interessati:

1. debug dmvpn condition peer NBMA x.x.x.x (dove x.x.x.x è l'indirizzo IP del dispositivo remoto).
2. debug dmvpn all: questo comando abilita i comandi di debug ISAKMP, IKEv2, IPSEC, DMVPN e NHRP.



Suggerimento: si consiglia di utilizzare il comando peer condition ogni volta che si abilitano i debug in modo da poter visualizzare la negoziazione del tunnel specifico.

Per visualizzare il flusso NHRP completo, sono stati utilizzati i comandi di debug successivi su ciascun dispositivo:

Raggio1

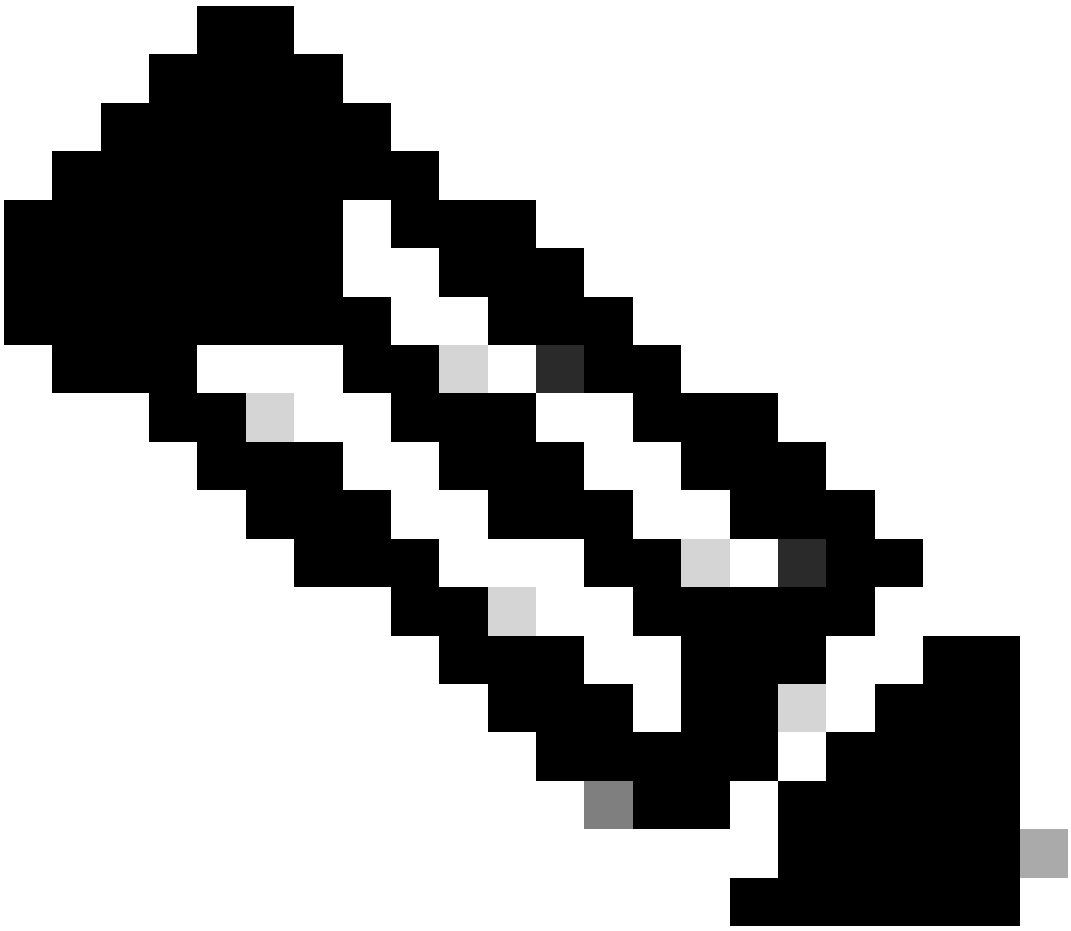
```
debug dmvpn condition peer NBMA 172.22.200.2
debug dmvpn condition peer NBMA 172.20.10.10
debug dmvpn all all
```

HUB

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

Raggio2

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



Nota: i debug devono essere abilitati e raccolti contemporaneamente su tutti i dispositivi

---

interessati.

I debug abilitati su tutti i dispositivi vengono visualizzati con il comando show debug:

<#root>

```
ROUTER#
show debug

IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:


Packet Infra debugs:

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

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

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

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

Dopo aver raccolto tutti i debug, è necessario avviare l'analisi dei debug sul spoke di origine (Spoke1), in modo da poter tracciare la negoziazione dall'inizio.

Output debug Spoke1:

<#root>

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

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

Old State = IKE_QM_R_QM2 New State = IKE_QM_PHASE2_COMPLETE


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

tunnel_protection_socket_up


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

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


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

reqid: 10


*Feb 1 01:31:36.429:

src NBMA: 172.21.100.1


*Feb 1 01:31:36.429:

src protocol: 10.10.10.1, dst protocol: 10.10.10.2


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

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

data:DMVPN

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

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

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

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

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

reqid: 10

*Feb 1 01:31:39.817:

src NBMA: 172.21.100.1

*Feb 1 01:31:39.817:

src protocol: 10.10.10.1, dst protocol: 10.10.10.2

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

data:DMVPN

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

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

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

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

Una volta avviato il processo NHRP Spoke1, i registri mostrano che il dispositivo sta inviando la richiesta di risoluzione NHRP. Il pacchetto ha alcune informazioni importanti, come i protocolli src NMBA e src che sono l'indirizzo IP NBMA e l'indirizzo IP del tunnel dell'origine spoke (Spoke1). È inoltre possibile visualizzare il valore del protocollo dst con l'indirizzo IP del tunnel di destinazione spoke (Spoke2). Ciò indica che Spoke1 richiede l'indirizzo NBMA di Spoke2 per completare il mapping. Anche sul pacchetto, è possibile trovare il valore richiesto che può aiutare a tenere traccia del pacchetto lungo il percorso. Questo valore rimarrà lo stesso durante l'intero processo e può essere utile per tenere traccia di un flusso specifico della negoziazione NHRP. Il pacchetto ha altri valori importanti per la negoziazione, come la stringa di autenticazione NHRP.

Dopo che il dispositivo ha inviato la richiesta di risoluzione NHRP, i registri mostrano che è stata inviata una ritrasmissione. Questo accade perché il dispositivo non vede la risposta di risoluzione NHRP e quindi invia nuovamente il pacchetto. Poiché Spoke1 non visualizza la risposta, è necessario tenere traccia del pacchetto sul dispositivo successivo nel percorso, ovvero l'HUB.

Output debug HUB:

<#root>

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

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

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

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

*Feb 1 01:31:34.263: (C-1) code: no error(0), flags: none
*Feb 1 01:31:34.263: prefix: 0, mtu: 9976, hd_time: 600
*Feb 1 01:31:34.263: addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 0, pref: 255
*Feb 1 01:31:34.263: Responder Address Extension(3):
*Feb 1 01:31:34.263: Forward Transit NHS Record Extension(4):
*Feb 1 01:31:34.263: Reverse Transit NHS Record Extension(5):
*Feb 1 01:31:34.263: Authentication Extension(7):
*Feb 1 01:31:34.263: type:Cleartext(1), data:DMVPN
*Feb 1 01:31:34.263: NAT address Extension(9):
*Feb 1 01:31:34.263: NHRP-DETAIL: netid_in = 10, to_us = 0
*Feb 1 01:31:34.263: NHRP-DETAIL:
```

Resolution request for afn 1 received on interface Tunnel10

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

Route lookup for destination 10.10.10.2

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

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

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

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

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

reqid: 10

*Feb 1 01:31:34.264:

src NBMA: 172.21.100.1

*Feb 1 01:31:34.264:

src protocol: 10.10.10.1, dst protocol: 10.10.10.2

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

code: no error(0)

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

client NBMA: 172.20.10.10

*Feb 1 01:31:34.264:

client protocol: 10.10.10.10

*Feb 1 01:31:34.264: Reverse Transit NHS Record Extension(5):

```
*Feb 1 01:31:34.264: Authentication Extension(7):
*Feb 1 01:31:34.264: type:Cleartext(1),
```

data:DMVPN

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

Utilizzando il valore della richiesta, è possibile osservare che l'HUB riceve la richiesta di risoluzione inviata da Spoke1. Nel pacchetto, i valori di src NBMA e src protocol sono le informazioni di Spoke1, e il valore di dst protocol è l'IP del tunnel di Spoke2, come è stato visto nei debug di Spoke1. Quando l'HUB riceve la richiesta di risoluzione, esegue una ricerca del percorso e inoltra il pacchetto a Spoke2. Nel pacchetto inoltrato, l'HUB aggiunge un'estensione contenente le proprie informazioni (indirizzo IP NBMA e indirizzo IP del tunnel).

I debug precedenti indicano che l'HUB sta inoltrando correttamente la richiesta di risoluzione a spoke 2. Pertanto, il passaggio successivo consiste nel confermare che Spoke2 lo sta ricevendo, elaborandolo correttamente e inviando a Spoke1 la risposta di risoluzione.

Output debug Spoke2:

<#root>

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

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

Old State = IKE_QM_IPSEC_INSTALL_AWAIT New State = IKE_QM_PHASE2_COMPLETE

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

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

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

NHRP:

Request was to us. Process the NHRP Resolution Request.

```
*Feb 1 01:31:34.648: NHRP-DETAIL: Multipath IP route lookup for 10.10.10.2 in vrf: global(0x0) yielded
*Feb 1 01:31:34.648: NHRP: nhrp_rtlookup for 10.10.10.2 in vrf: global(0x0) yielded interface Tunnel10,
*Feb 1 01:31:34.648: NHRP: Request was to us, responding with ouraddress
*Feb 1 01:31:34.648: NHRP: Checking for delayed event 10.10.10.1/10.10.10.2 on list (Tunnel10 vrf: glob
*Feb 1 01:31:34.648: NHRP: No delayed event node found.
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10: Checking to see if we need to delay for src 172.22.200.2 dst
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10: crypto_ss_listen_start already listening
```

```
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Opening a socket with profile IPSE
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F1
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Socket is already open. Ignoring.
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F1
*Feb 1 01:31:34.648: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): tunnel is already open!
*Feb 1 01:31:34.648: NHRP: No need to delay processing of resolution event NBMA src:172.22.200.2 NBMA d
*Feb 1 01:31:34.648: NHRP-MEF: No vendor private extension in NHRP packet
*Feb 1 01:31:34.649: NHRP-CACHE: Tunnel10: Cache update for target 10.10.10.1/32 vrf: global(0x0) label
*Feb 1 01:31:34.649: 172.21.100.1 (flags:0x2080)
*Feb 1 01:31:34.649: NHRP:
```

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

```
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10: crypto_ss_listen_start already listening
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Opening a socket with profile IPSE
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): connection lookup returned 80007F1
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Found an existing tunnel endpoint
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): tunnel_protection_stop_pending_tim
*Feb 1 01:31:34.649: IPSEC-IFC MGRE/Tu10(172.22.200.2/172.21.100.1): Socket is already open. Ignoring.
*Feb 1 01:31:34.653:
```

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

```
*Feb 1 01:31:34.653: NHRP: Peer capability:0
*Feb 1 01:31:34.653: NHRP-CACHE: Inserted subblock node(1 now) for cache: Target 10.10.10.1/32 nhop 10.
*Feb 1 01:31:34.653: NHRP-CACHE: Converted internal dynamic cache entry for 10.10.10.1/32 interface Tun
*Feb 1 01:31:34.653: NHRP-EVE: NHP-UP: 10.10.10.1, NBMA: 172.21.100.1
*Feb 1 01:31:34.653: NHRP-MEF: No vendor private extension in NHRP packet
*Feb 1 01:31:34.653: NHRP-CACHE: Tunnel10: Internal Cache add for target 10.10.10.2/32 vrf: global(0x0)
*Feb 1 01:31:34.653: 172.22.200.2 (flags:0x20)
*Feb 1 01:31:34.653: NHRP: Attempting to send packet through interface Tunnel10 via DEST dst 10.10.10.1
*Feb 1 01:31:34.654: NHRP-DETAIL: First hop route lookup for 10.10.10.1 yielded 10.10.10.1, Tunnel10
*Feb 1 01:31:34.654:
```

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

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

reqid: 10

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

src NBMA: 172.21.100.1

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

src protocol: 10.10.10.1, dst protocol: 10.10.10.2

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

client NBMA: 172.22.200.2

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


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


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


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


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


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


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

La richiesta corrisponde al valore visualizzato negli output precedenti, pertanto viene confermato che il pacchetto di richiesta di risoluzione NHRP inviato da Spoke1 raggiunge Spoke2. Questo pacchetto attiva una ricerca route in Spoke2 e si rende conto che la richiesta di risoluzione è per se stessa, pertanto Spoke2 aggiunge le informazioni da Spoke1 alla relativa tabella NHRP. Prima di inviare il pacchetto di risposta alla risoluzione a Spoke1, il dispositivo aggiunge le proprie informazioni (indirizzo IP NBMA e indirizzo IP tunnel) in modo che Spoke1 possa utilizzare tale pacchetto per aggiungere tali informazioni al proprio database.

In base a tutti i debug rilevati, la risposta di risoluzione NHRP inviata da Spoke2 non arriva a Spoke1. È possibile eliminare l'HUB dal problema durante la ricezione e l'inoltro del pacchetto NHRP Resolution Request, come previsto. Pertanto, il passo successivo è catturare immagini tra Spoke1 e Spoke2 per ottenere maggiori dettagli sul problema.

Embedded Packet Capture

La funzione di acquisizione dei pacchetti integrata consente di analizzare il traffico che attraversa il dispositivo. Il primo passaggio per configurarlo è la creazione di un elenco degli accessi che includa il traffico che si desidera catturare su entrambi i flussi (in entrata e in uscita).

Per questo scenario, vengono utilizzati gli indirizzi IP NBMA:

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

Quindi, configurare l'acquisizione utilizzando il comando monitor capture <NOME_ACQUISIZIONE> access-list <NOME_ACL> buffer size 10 interface <INTERFACCIA_WAN> sia che avviare l'acquisizione con il comando monitor capture <NOME_ACQUISIZIONE> .

Acquisire la configurazione su Spoke1 e Spoke2:

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

Per visualizzare l'output dell'acquisizione, usare il comando show monitor capture <NOME_ACQUISIZIONE> buffer brief.

Acquisizione output 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
```

Acquisizione output Spoke2:

<#root>

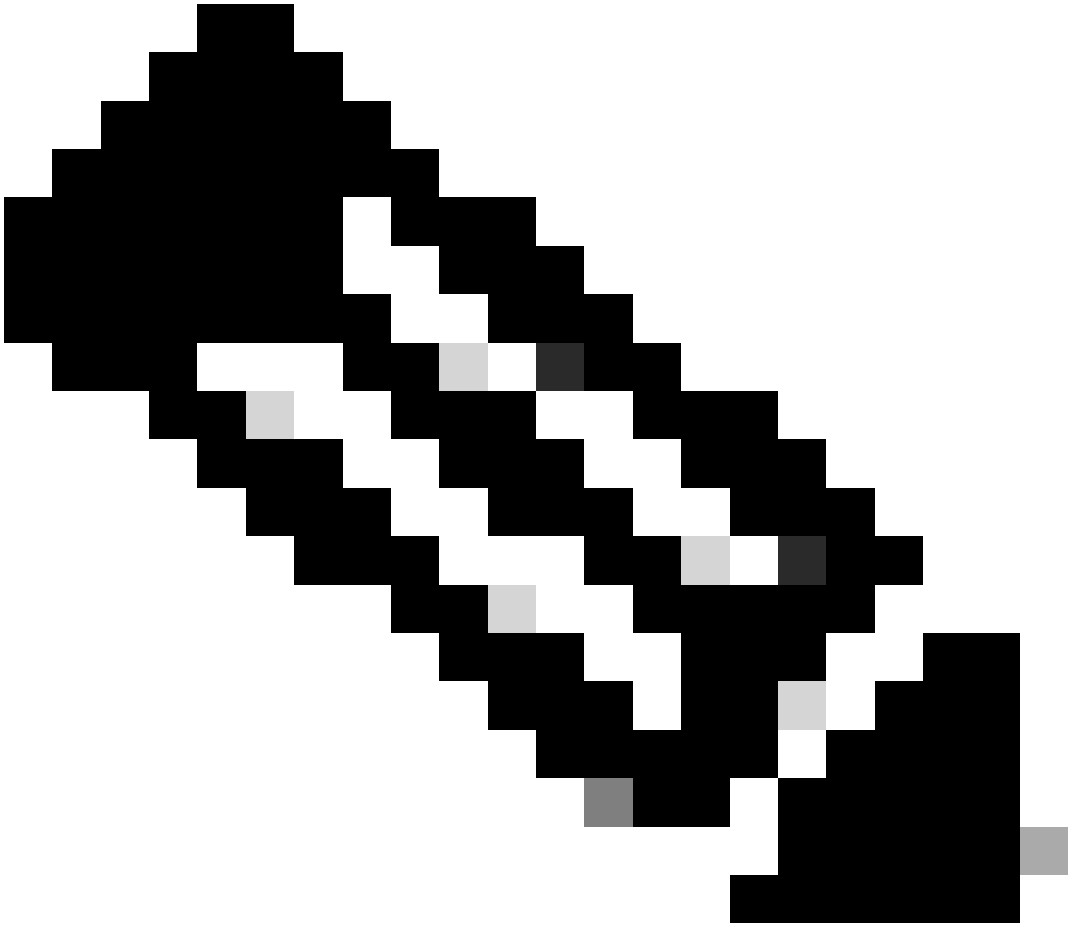
```
SPOKE2#show monitor capture CAP buffer brief
------------------------------------------------------------------------
#   size   timestamp     source             destination      dscp    protocol
------------------------------------------------------------------------
   0  210    0.000000   172.22.200.2    ->  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
```

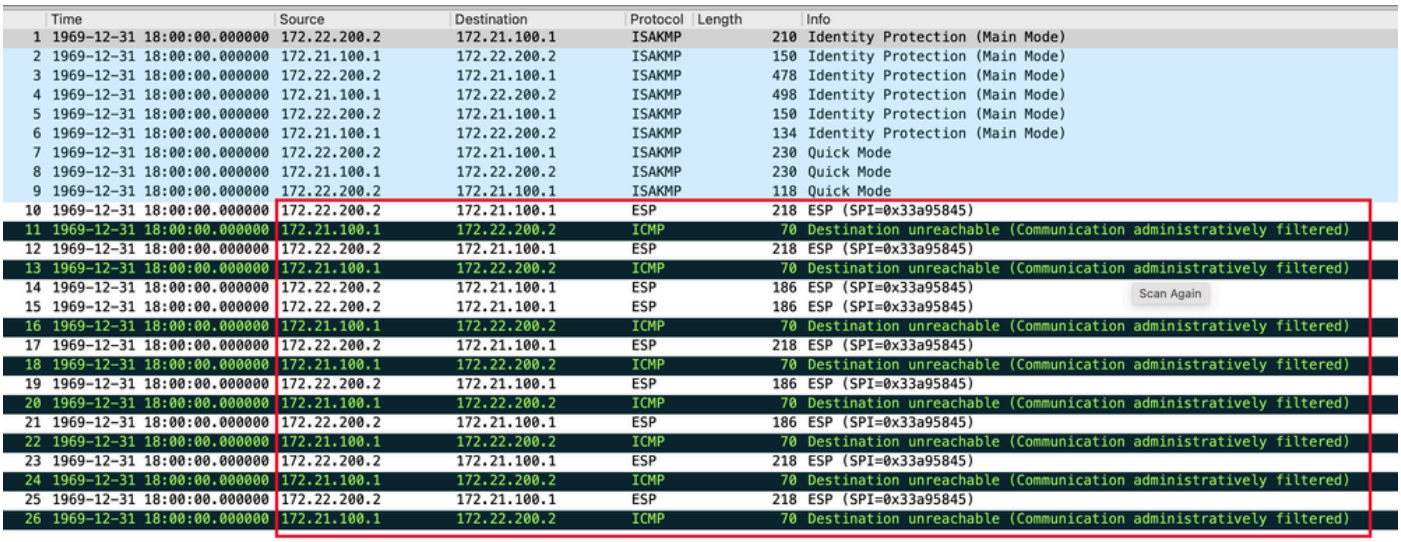
L'output delle acquisizioni mostra che i pacchetti iniziali sono traffico UDP, indicando la negoziazione IKE/IPSEC. In seguito, Spoke2 invia la risposta di risoluzione a Spoke1, che è visto come traffico ESP (pacchetto 9). Dopo questa operazione, il flusso del traffico previsto è ESP, ma il pacchetto successivo visualizzato è il traffico ICMP da Spoke1 a Spoke2.

Per analizzare il pacchetto in modo più approfondito, è possibile esportare il file pcap dal dispositivo eseguendo il comando show monitor capture <CAPTURE_NAME> buffer dump. Quindi usate uno strumento di decodifica per convertire l'output di dump in un file pcap in modo da poterlo aprire con Wireshark.



Nota: Cisco dispone di un analizzatore di pacchetti dove è possibile trovare configurazione di acquisizione, esempi e un decoder: Cisco TAC Tool - Packet Capture Config Generator e Analyzer

Output Wireshark:

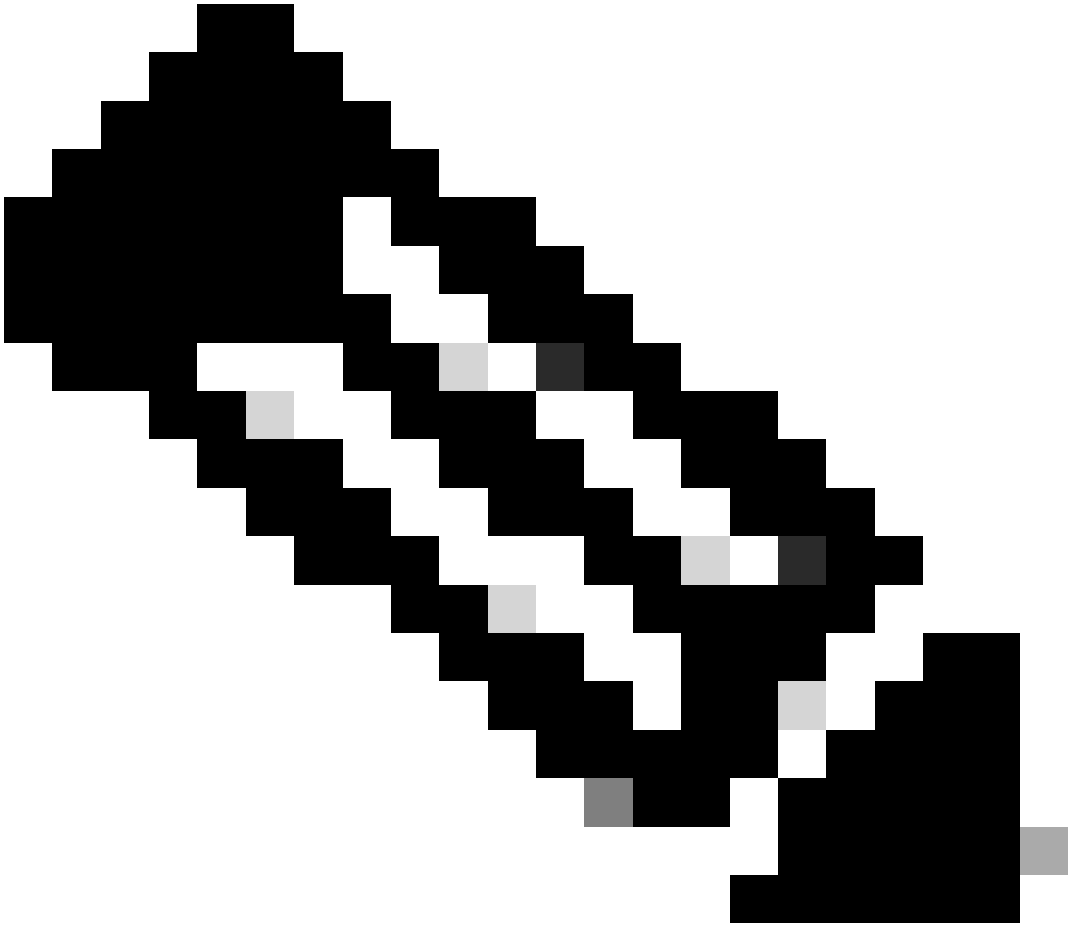


	Time	Source	Destination	Protocol	Length	Info
1	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP	210	Identity Protection (Main Mode)
2	1969-12-31 18:00:00.000000	172.21.100.1	172.22.200.2	ISAKMP	150	Identity Protection (Main Mode)
3	1969-12-31 18:00:00.000000	172.22.200.2	172.21.100.1	ISAKMP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

Scan Again

Cattura output su Wireshark

Il contenuto del pacchetto ICMP visualizza il messaggio di errore Destination unreachable (Destinazione irraggiungibile) (Comunicazione filtrata manualmente). Ciò indica che esiste un filtro, ad esempio un ACL del router o un firewall, che influenza il traffico lungo il percorso. Nella maggior parte dei casi, il filtro è configurato sul dispositivo che invia il pacchetto (in questo caso, Spoke1), ma anche i dispositivi intermedi possono inviarlo.



Nota: l'output di Wireshark è lo stesso su entrambi i raggi.

Funzione Cisco IOS® XE Datapath Packet Trace

La funzionalità di traccia dei pacchetti del percorso dati Cisco IOS XE viene utilizzata per analizzare la modalità di elaborazione del traffico da parte del dispositivo. Per configurarlo, è necessario creare un elenco degli accessi contenente il traffico che si desidera acquisire sia sul traffico in entrata che in uscita.

Per questo scenario, vengono utilizzati gli indirizzi IP NBMA.

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

Configurare quindi la funzionalità di analisi dei file e impostare le condizioni di debug per l'utilizzo dell'elenco degli accessi. Infine, avviare la condizione.

```
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: abilita la traccia fia dettagliata, arrestandola dopo l'acquisizione della quantità di pacchetti configurati
- debug platform condition ipv4 access-list <ACL-NAME> both: imposta una condizione sul dispositivo utilizzando l'elenco degli accessi configurato in precedenza
- debug platform condition start: avvia la condizione

Per esaminare l'output della traccia finale, utilizzare i comandi successivi.

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

Spoke1 show platform packet-trace: output statistico:

<#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
```

Nell'output show platform packet-trace statistics, è possibile visualizzare i contatori dei pacchetti elaborati dal dispositivo. Ciò consente di visualizzare i pacchetti in entrata e in uscita e di controllare se il dispositivo sta scartando alcuni pacchetti, insieme al motivo della perdita.

Nell'output mostrato, Spoke1 sta scartando alcuni pacchetti con la descrizione Ipv4Acl. Per analizzare ulteriormente questi pacchetti, è possibile usare il comando show platform packet-trace summary.

Spoke1 show platform packet-trace: output di riepilogo:

<#root>

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

9      Gi1                       Gi1                       DROP   8   (Ipv4Acl)


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

12     Gi1                       Gi1                       DROP   8   (Ipv4Acl)


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

15     Gi1                       Gi1                       DROP   8   (Ipv4Acl)
```

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

Con questo output, è possibile visualizzare tutti i pacchetti in arrivo e in uscita dal dispositivo, nonché le interfacce in entrata e in uscita. Viene visualizzato anche lo stato del pacchetto, per indicare se è stato inoltrato, scartato o elaborato internamente (punt).

In questo esempio, l'output mostrato di seguito è utile per identificare i pacchetti scartati dal dispositivo. Il comando show platform packet-trace packet <PACKET_NUMBER> permette di verificare il modo in cui il dispositivo elabora il pacchetto.

Spoke1 show platform packet-trace pacchetto <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>

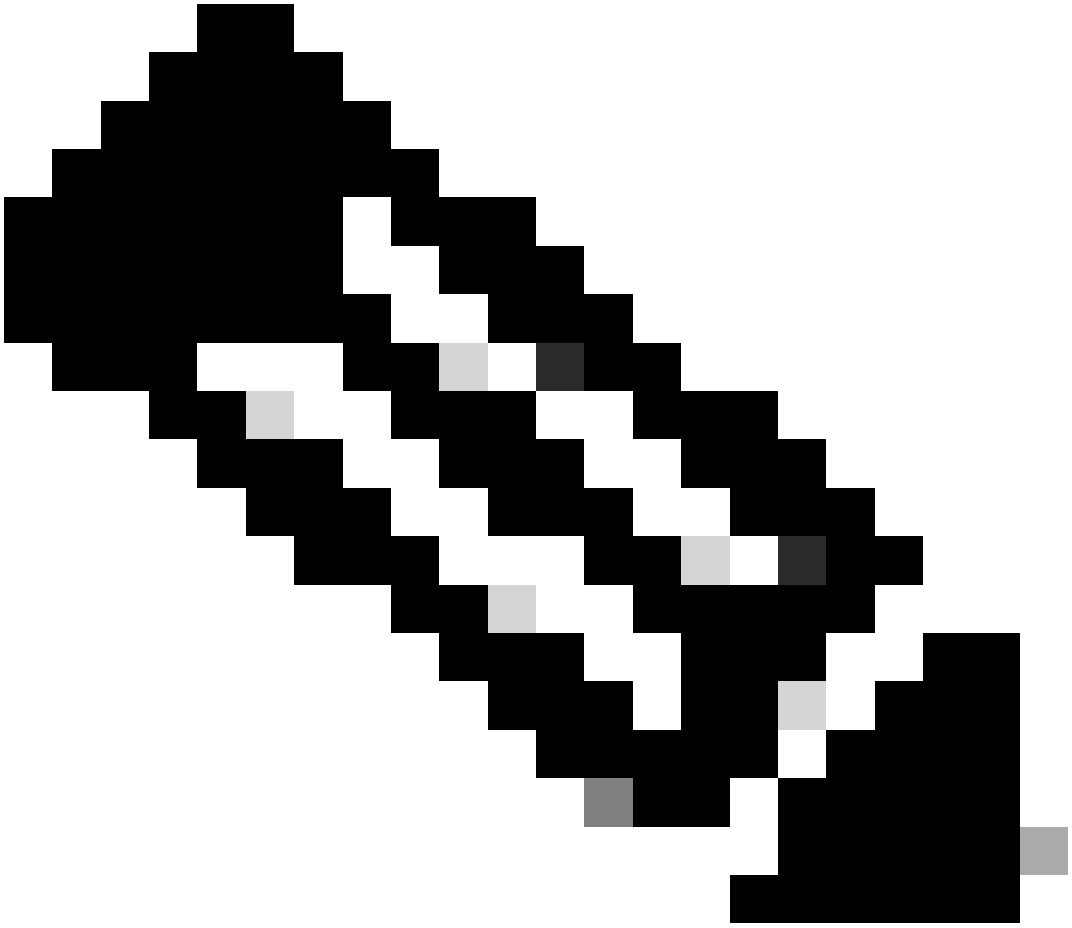
Lapsed time : 174 ns
Feature: INPUT_DROP

```
     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
```

Nella prima parte, è possibile vedere l'interfaccia in entrata e in uscita e lo stato del pacchetto. A questo punto, segue la seconda parte dell'output, in cui è possibile trovare gli indirizzi IP di origine e di destinazione e il protocollo.

In ciascuna fase successiva viene mostrato come il dispositivo elabora questo particolare pacchetto. Questo documento offre informazioni dettagliate su configurazioni come NAT (Network Address Translation) o un elenco degli accessi o altri fattori che potrebbero influire su di esso.

In questo caso, è possibile identificare il protocollo del pacchetto come ESP, l'IP di origine come indirizzo IP NBMA di Spoke2 e l'IP di destinazione come indirizzo IP NBMA di Spoke1. Ciò indica che si tratta del pacchetto mancante nella negoziazione NHRP. Inoltre, si osserva che nessuna interfaccia in uscita viene specificata in alcuna fase, indicando che qualcosa ha influenzato il traffico prima che potesse essere inoltrato. Nella penultima fase, è possibile vedere che il dispositivo sta riducendo il traffico in entrata sull'interfaccia specificata (Gigabit Ethernet1). Nell'ultima fase viene mostrato un elenco degli accessi all'input, in cui si suggerisce che potrebbe esserci una configurazione sull'interfaccia che causa il rilascio.



Nota: se dopo aver utilizzato tutti gli strumenti di risoluzione dei problemi elencati in questo documento, gli spoke coinvolti nella negoziazione non mostrano alcun segno di diminuzione o di impatto sul traffico, ciò conclude la risoluzione dei problemi su tali dispositivi.

Il passaggio successivo deve essere il controllo dei dispositivi intermedi, quali firewall, switch e ISP.

Soluzione

In questo caso, il passaggio successivo è controllare l'interfaccia mostrata negli output precedenti. In questo caso, è necessario controllare la configurazione per verificare se vi sono elementi che influiscono sul traffico.

Configurazione interfaccia WAN:

<#root>

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

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

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

Come parte della sua configurazione, all'interfaccia è applicato un gruppo di accesso. È importante verificare che gli host configurati nell'elenco degli accessi non interferiscano con il traffico utilizzato per la negoziazione NHRP.

<#root>

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

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

La seconda istruzione dell'elenco degli accessi nega la comunicazione tra l'indirizzo IP NBMA di Spoke2 e l'indirizzo IP NBMA di Spoke1, causando la perdita rilevata in precedenza. Dopo aver rimosso il gruppo di accesso dall'interfaccia, la comunicazione tra i due raggi ha esito positivo:

```
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
```

Il tunnel IPSEC è attivo e ora mostra incapsulamenti e decapsulamenti su entrambi i dispositivi:

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:
```

Raggio2:

<#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:
```

La tabella DMVPN di Spoke1 mostra ora il mapping corretto su entrambe le voci:

<#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
```

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).