排除DMVPN第2阶段分支到分支隧道故障

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

理论背景

拓扑

故障排除步骤

- 初始验证
- 故障排除工具
  - 有用的命令
  - 调试
  - 嵌入式数据包捕获
  - Cisco IOS® XE数据路径数据包跟踪功能
- 解决方案

简介

本文档介绍如何在阶段2分支到分支DMVPN隧道未建立时对其进行故障排除。

先决条件

要求

思科建议您了解以下主题：

- 动态多点虚拟专用网络(DMVPN)
- IKE/IPSEC协议
- 下一跳解析协议 (NHRP)

使用的组件

本文档基于以下软件版本：

- 思科CSR1000V (VXE) -版本17.03.08

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

本文档介绍如何针对常见的DMVPN问题配置和使用不同的故障排除工具。 问题在于第2阶段DMVPN隧道协商失败，在该阶段，源分支，DMVPN状态显示为与到目标分支的正确非广播多路访问(NBMA)/隧道映射的UP。但是，在目标分支上显示不正确的映射。

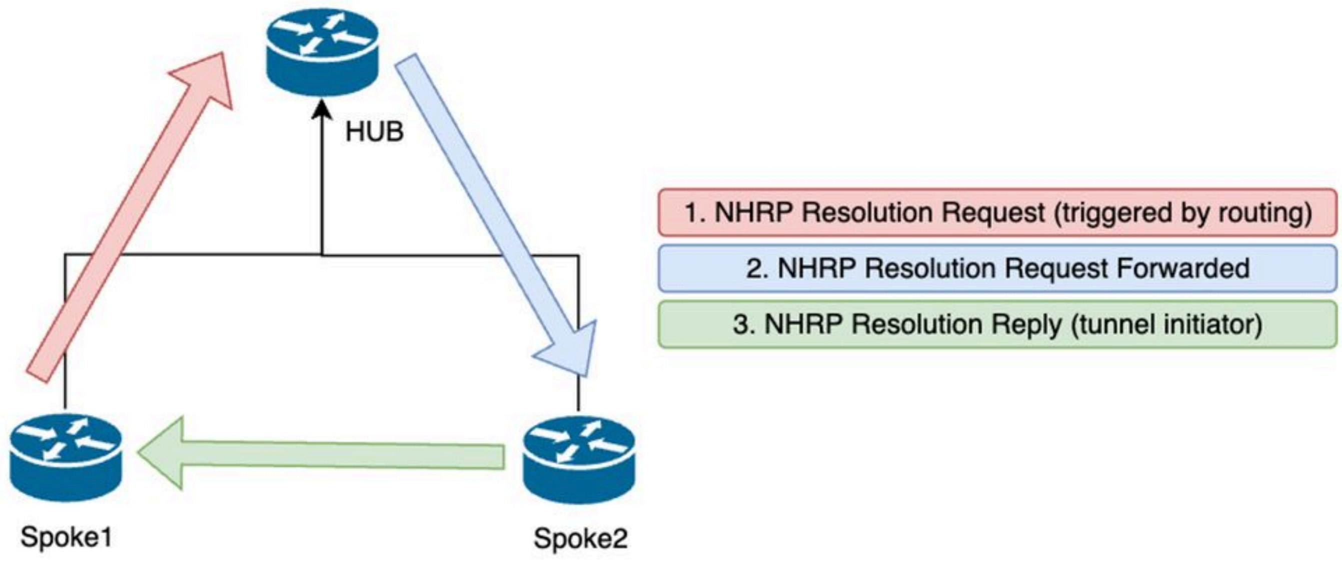
理论背景

在设置DMVPN第2阶段时，必须了解如何建立分支到分支隧道。本部分简要概述此阶段的NHRP流程。

在DMVPN第2阶段，您可以按需构建动态分支到分支隧道。 这可能是因为，在DMVPN云（集线器和分支）中的所有设备上，隧道接口的模式更改为通用路由封装(GRE)多点。此阶段的一个关键特征是，其他设备不会将集线器视为下一跳。相反，所有分支都有彼此的路由信息。 在第2阶段建立分支到分支隧道时，会触发NHRP进程，其中分支获取有关其他分支的信息，并在NBMA和隧道IP地址之间进行映射。

接下来的步骤将列出如何触发NHRP解析过程：

1. 当源分支尝试到达目标分支的LAN时，它会执行路由查找，触发解析请求消息，以获取目标分支的NBMA地址。源分支将此初始消息发送到中心。
2. 中心点接收解析请求并将其转发到目标分支。
3. 目标分支向源分支发送解析应答。如果隧道配置链接了IPSEC配置文件：
    - NHRP解析过程被延迟到IKE/IPSEC协议可以建立为止。
    - 目标分支发起并建立IKE/IPSEC隧道。
    - 然后，NHRP进程恢复，并且目标分支使用IPSEC隧道作为传输方法向源分支发送解析应答。

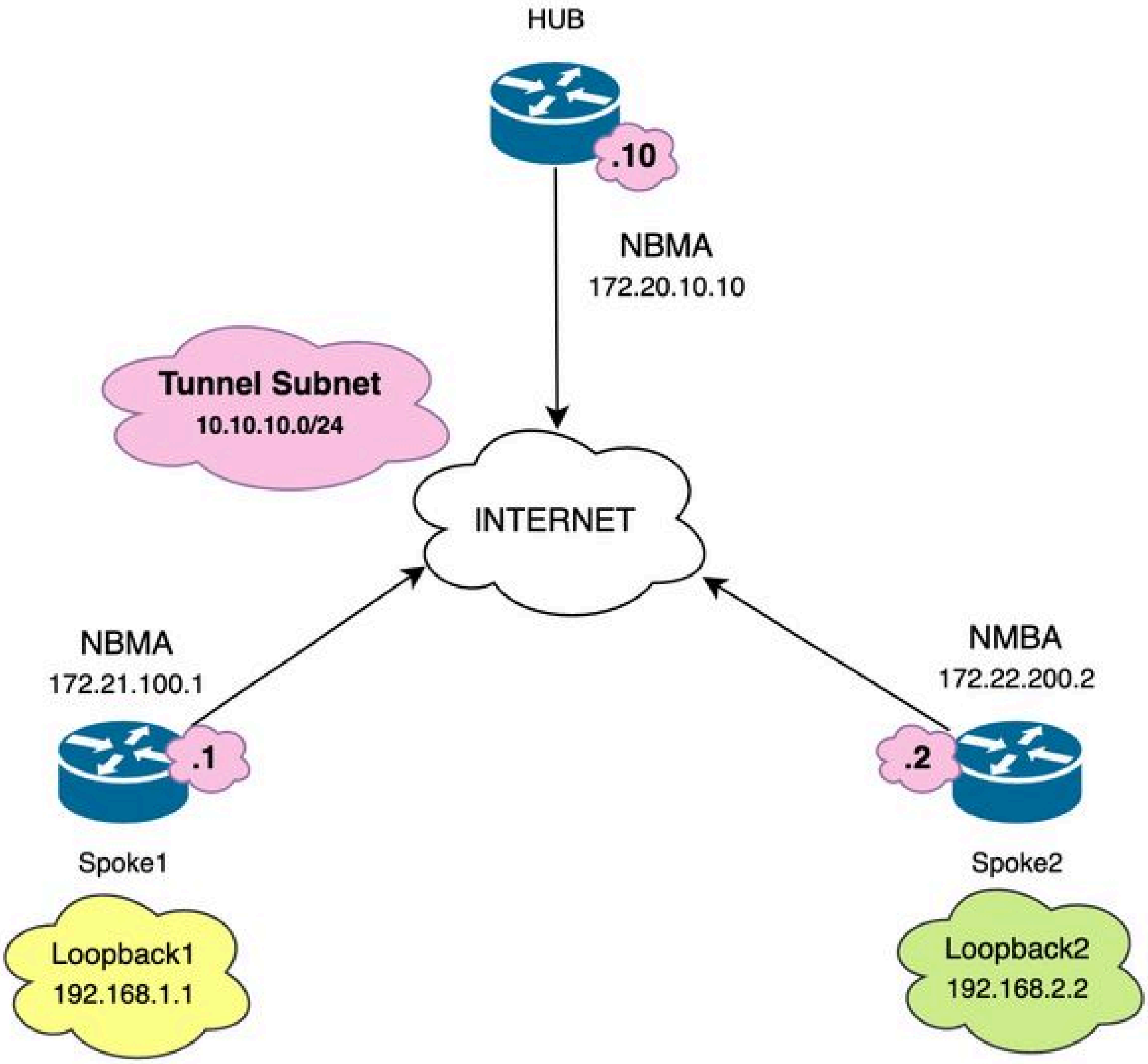

HUB
1. NHRP Resolution Request (triggered by routing)
2. NHRP Resolution Request Forwarded
3. NHRP Resolution Reply (tunnel initiator)
Spoke1
Spoke2

第2阶段分支之间的NHRP消息流

注意：在开始解析过程之前，所有辐射点都必须已经注册到HUB。

拓扑

下图显示用于该场景的拓扑：



使用的网络图和IP子网

故障排除步骤

在此场景中，Spoke1和Spoke2之间未建立分支到分支隧道，这会影响其本地资源（由环回接口表示）之间的通信，因为它们无法互相访问。

```
SPOKE1#ping 192.168.2.2 source loopback1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

初始验证

遇到这种情况时，首先必须验证隧道配置，并确保两台设备在配置中具有正确的值。要查看隧道配置，请运行命令show running-config interface tunnel<ID>。

分支1隧道配置：

<#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
```

分支2隧道配置：

<#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
```

在配置中，您需要验证到HUB的映射是否正确，设备之间的NHRP身份验证字符串是否匹配，两个分支配置了相同的DMVPN阶段，并且如果使用IPSEC保护，请验证是否应用了正确的加密配置。

如果配置正确并且包含IPSEC保护，则需要验证IKE和IPSEC协议是否正常工作。这是因为NHRP使用IPSEC隧道作为传输方法来完全协商。要验证IKE/IPSEC协议的状态，请运行show crypto IPSEC sa peer x.x.x.x命令（其中x.x.x.x是尝试建立隧道的分支的NBMA IP地址）。

注意：要验证IPSEC隧道是否已开启，入站和出站封装安全负载(ESP)部分必须具有隧道信息（SPI、转换集等）。此部分中显示的所有值在两端必须匹配。

注意：如果发现IKE/IPSEC存在任何问题，则故障排除必须重点关注这些协议。

Spoke1上的IKE/IPSEC隧道状态：

<#root>

SPOKE1#

show crypto IPSEC sa peer 172.22.200.2

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

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

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


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


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

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


inbound esp sas:



spi: 0x84502A19(2219846169)



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

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

inbound ah sas:

inbound pcp sas:


outbound esp sas:



spi: 0x6F6BF94A(1869347146)



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

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

outbound ah sas:

outbound pcp sas:
```

Spoke2上的IKE/IPSEC隧道状态：

<#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:
```

输出显示，在两个分支上，IPSEC隧道均处于启用状态，但是，Spoke2显示加密数据包（封装）但没有解密数据包（解密）。同时，Spoke1不会显示流经IPSEC隧道的任何数据包。这表示问题可能出在NHRP协议上。

故障排除工具

执行初始验证并确认配置和IKE/IPSEC协议（如果需要）未导致通信问题后，您可以使用本节中介绍的工具继续故障排除。

有用的命令

命令show dmvpn interface tunnel<ID>可以提供DMVPN特定会话信息(NBMA/隧道IP地址、隧道状态、打开/关闭时间和属性)。可以使用detail关键字显示来自加密会话/套接字的详细信息。必须说明隧道两端状态必须匹配。

分支1 show dmvpn interface tunnel<ID>输出：

<#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
```

分支2 show dmvpn interface tunnel<ID>输出：

<#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
```

每台设备上的输出显示每个分支的不同信息。在Spoke1表中，您可以看到Spoke 2的条目不包含正确的NBMA IP地址，并且属性显示为不完整(I2)。另一方面，Spoke2表显示正确的映射（NBMA/隧道IP地址）和状态up，表示隧道已完全协商。

在故障排除过程中，以下命令很有用：

- show ip nhrp：显示NHRP映射信息
- show ip nhrp traffic interface tunnel10：显示NHRP流量统计信息

注意：有关命令规范（语法、说明、关键字、示例），请参阅《命令参考：Cisco IOS安全命令参考：命令S至Z》

调试

在验证之前的信息并确认隧道遇到协商问题后，必须启用调试以观察如何交换NHRP数据包。必须在所有相关设备上启用下一次调试：

1. debug dmvpn condition peer NBMA x.x.x.x（其中x.x.x.x是远程设备IP地址）。
2. debug dmvpn all：此命令启用ISAKMP、IKEv2、IPSEC、DMVPN和NHRP调试命令。

提示：建议每次启用调试时都使用peer condition命令，以便您可以查看特定隧道的协商。

要查看完整的NHRP流程，请在每台设备上使用下一条debugs命令：

分支1

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

集线器

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

分支2

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

注意：必须在涉及的所有设备上同时启用和收集调试。

所有设备上启用的调试都通过命令show debug显示：

<#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
```

收集所有调试后，您必须开始分析源分支(Spoke1)上的调试，这样您就可以从头开始跟踪协商。

Spoke1调试输出：

<#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
```

一旦Spoke1 NHRP进程开始，日志显示设备正在发送NHRP解析请求。数据包包含一些重要信息，如src NMBA和src protocol，它们是源分支(Spoke1)的NBMA IP地址和隧道IP地址。您还可以看到具有目标分支(Spoke2)的隧道IP地址的dst protocol值。这表示Spoke1要求Spoke2的NBMA地址

完成映射。 此外，在数据包上，可以找到reqid值，该值可帮助您跟踪路径中的数据包。此值在整个过程中将保持不变，并且有助于跟踪NHRP协商的特定流程。数据包具有另一个对协商至关重要的值，如NHRP身份验证字符串。

设备发送NHRP解析请求后，日志显示已发送重新传输。这是因为设备未看到NHRP解析响应，因此它再次发送数据包。由于Spoke1看不到响应，因此有必要在路径中的下一台设备（即HUB）上跟踪该数据包。

集线器调试输出：

<#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: 172.22.200
*Feb 1 01:31:34.265: NHRP: 129 bytes out Tunnel10
```

使用reqid值，您可以观察到，HUB收到了Spoke1发送的分辨率请求。在数据包中，src NBMA和src protocol的值是来自Spoke1的信息，而dst protocol的值是Spoke2的隧道IP，正如在Spoke1的调试中看到的那样。中心路由器收到解析请求后，会执行路由查找并将数据包转发到Spoke2。在转发的数据包中，集线器添加一个包含其自身信息（NBMA IP地址和隧道IP地址）的扩展。

前面的调试显示，HUB正将解析请求正确转发到spoke 2。因此，下一步是确认Spoke2正在接收解析应答，并对其进行正确处理，然后向Spoke1发送解析应答。

Spoke2调试输出：

<#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 Tunn
*Feb 1 01:31:34.653: NHRP-EVE: NHP-UP: 10.10.10.1, NBMA: 172.21.100.1
*Feb 1 01:31:34.653: NHRP-MEF: No vendor private extension in NHRP packet
*Feb 1 01:31:34.653: NHRP-CACHE: Tunnel10: Internal Cache add for target 10.10.10.2/32 vrf: global(0x0)
*Feb 1 01:31:34.653: 172.22.200.2 (flags:0x20)
*Feb 1 01:31:34.653: NHRP: Attempting to send packet through interface Tunnel10 via DEST dst 10.10.10.1
*Feb 1 01:31:34.654: NHRP-DETAIL: First hop route lookup for 10.10.10.1 yielded 10.10.10.1, Tunnel10
*Feb 1 01:31:34.654:

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


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

 reqid: 10


*Feb 1 01:31:34.654:

src NBMA: 172.21.100.1


*Feb 1 01:31:34.654:

src protocol: 10.10.10.1, dst protocol: 10.10.10.2


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

client NBMA: 172.22.200.2


*Feb 1 01:31:34.654:

client protocol: 10.10.10.2


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

client NBMA: 172.22.200.2


*Feb 1 01:31:34.654:

client protocol: 10.10.10.2
```

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

client NBMA: 172.20.10.10

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

client protocol: 10.10.10.10

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

data:DMVPN

```
*Feb 1 01:31:34.655: NAT address Extension(9):
*Feb 1 01:31:34.655: NHRP: Encapsulation succeeded. Sending NHRP Control Packet NBMA Address: 172.21.100
*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 80007F10
*Feb 1 01:31:34.655: NHRP-DETAIL: Deleted delayed event on interfaceTunnel10 dest: 172.21.100.1
```

reqid与之前输出中看到的值匹配，因此确认Spoke1发送的NHRP解析请求数据包到达Spoke2。此数据包在Spoke2上触发路由查找，并意识到解析请求是为其本身提供的，因此Spoke2将来自Spoke1的信息添加到其NHRP表中。在将解析应答数据包发送回Spoke1之前，设备会添加自己的信息（NBMA IP地址和隧道IP地址），以便Spoke1可以使用该数据包将该信息添加到数据库。

根据显示的所有调试，从Spoke2发送的NHRP解析应答未到达Spoke1。由于集线器正在按预期接收和转发NHRP解析请求数据包，因此可以从此问题中丢弃集线器。因此，下一步是捕获Spoke1和Spoke2之间的数据以获得有关该问题的更多详细信息。

嵌入式数据包捕获

通过嵌入式数据包捕获功能，您可以分析通过设备的流量。配置它的第一步是创建一个访问列表，其中包括要在两个通信流（入站和出站）上捕获的流量。

对于此情况，使用NBMA IP地址：

```
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
```

然后，使用monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 interface <WAN_INTERFACE> both命令配置捕获，并使用命令monitor capture <CAPTURE_NAME> start开始捕获。

捕获Spoke1和Spoke2上的配置：

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

要显示捕获的输出，请使用命令show monitor capture <CAPTURE_NAME> buffer brief。

捕获输出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
```

捕获输出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
```

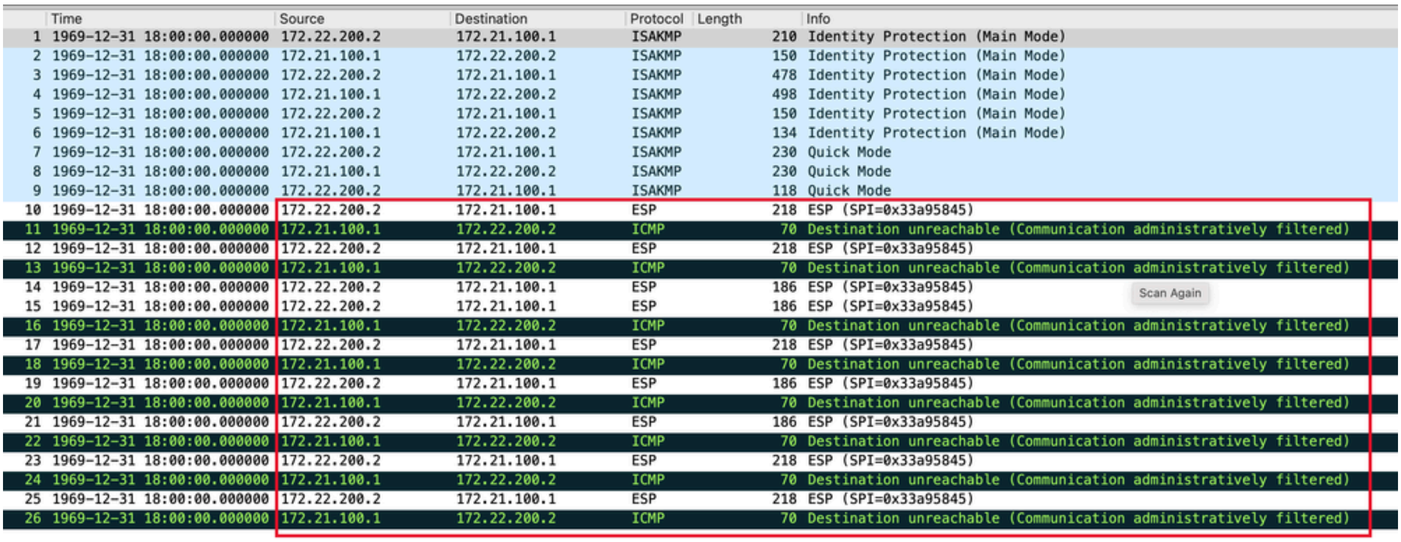
```
 18   70   26.859001   172.21.100.1    ->  172.22.200.2     0  BE   ICMP


 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
```

捕获的输出显示初始数据包是UDP流量，表示IKE/IPSEC协商。之后，Spoke2向Spoke1发送解析应答，该应答被视为ESP流量（数据包9）。在此之后，预期的流量为ESP，但是下一个数据包是从Spoke1到Spoke2的ICMP流量。

要深入分析数据包，可以通过运行命令show monitor capture <CAPTURE_NAME> buffer dump从设备导出pcap文件。然后使用解码器工具将转储输出转换为pcap文件，以便使用Wireshark打开该文件。

注意：思科拥有一个数据包分析器，您可以在其中找到捕获配置、示例和解码器：思科TAC工具-数据包捕获配置生成器和分析器

Wireshark输出：



```
    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)
```

捕获Wireshark上的输出

ICMP数据包的内容显示错误消息Destination unreachable (Communication administratively filtered)。这表示存在某种过滤器，例如影响路径中流量的路由器ACL或防火墙。大多数情况下，发送数据包的设备（本例中为Spoke1）上会配置过滤器，但中间设备也可以发送它。

注意：两个分支上的Wireshark输出相同。

Cisco IOS® XE数据路径数据包跟踪功能

Cisco IOS XE数据路径数据包跟踪功能用于分析设备如何处理流量。要配置它，您需要创建一个访问列表，包括要在两个通信流（入站和出站）上捕获的流量。

对于此情况，使用NBMA IP地址。

```
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
```

然后，配置fia-trace功能并设置调试条件以使用访问列表。最后，启动条件。

```
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：启用详细的fia跟踪，并在捕获配置的数据包数量后停止跟踪
- debug platform condition ipv4 access-list <ACL-NAME> both：使用之前配置的访问列表设置设备上的条件
- debug platform condition start：启动条件

要查看fia-trace的输出，请使用以下命令。

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

Spoke1 show platform packet-trace statistics输出：

<#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
```

在show platform packet-trace statistics输出中，您可以看到设备处理的数据包的计数器。这允许您查看入站和出站数据包，并检查设备是否丢弃任何数据包以及丢弃原因。

在显示的输出中，Spoke1正在丢弃描述为Ipv4Acl的一些数据包。要进一步分析这些数据包，可以使用命令show platform packet-trace summary。

Spoke1 show platform packet-trace summary输出：

<#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
```

通过此输出，您可以看到每个数据包到达和离开设备，以及入口和出口接口。还会显示数据包的状态，指明它是被转发、丢弃还是被内部处理（传送）。

在本例中，此输出有助于识别设备丢弃的数据包。使用命令show platform packet-trace packet <PACKET_NUMBER>，您可以看到设备如何处理该特定数据包。

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

<#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
```

在第一部分，您可以看到入口和出口接口以及数据包的状态。然后是输出的第二部分，您可以在其中查找源IP地址和目的IP地址以及协议。

后续的每个阶段都会显示设备如何处理此特定数据包。 这样可以深入了解网络地址转换(NAT)或访问列表等任何配置或可能影响这些配置的其它因素。

在这种情况下，可以确定数据包的协议是ESP，源IP是Spoke2的NBMA IP地址，目标IP是Spoke1的NBMA IP地址。这表示这是NHRP协商中缺少的数据包。此外，我们观察到，在任何阶段都没有指定出口接口，这表明有些内容在转发流量之前影响了流量。在倒数第二阶段，您可以看到设备正在丢弃指定接口(GigabitEthernet1)上的入站流量。最后阶段显示输入访问列表，表明接口上可能存在某种配置导致丢弃。

注意：在使用本文档中列出的所有故障排除工具后，如果协商中涉及的分支未显示任何丢弃或影响流量的迹象，则这些设备的故障排除到此结束。

下一步必须检查中间设备，例如防火墙、交换机和ISP。

解决方案

如果出现这种情况，下一步是检查前面输出中显示的接口。 这包括检查配置以验证是否存在影响流量的内容。

WAN 接口配置:

<#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
```

作为配置的一部分，接口应用了访问组。必须验证在访问列表中配置的主机不会干扰用于NHRP协商的流量。

<#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)
```

访问列表的第二条语句拒绝Spoke2的NBMA IP地址和Spoke1的NBMA IP地址之间的通信，导致之前看到的丢弃。从接口中删除访问组后，两个分支之间的通信成功：

```
SPOKE1#ping 192.168.2.2 source loopback1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/3 ms
```

IPSEC隧道处于启用状态，现在它显示两台设备上的封装和解封：

辐射点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:
```

分支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:
```

现在，Spoke1的DMVPN表在两个条目上显示正确的映射：

<#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
```

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。