Risoluzione dei problemi di rilevamento inoltro bidirezionale vEdge e di connessioni al piano dati

Sommario

Introduzione

In questo documento vengono descritti i problemi di connessione del piano dati vEdge dopo una connessione al piano di controllo, ma non la connettività del piano dati tra i siti.

Prerequisiti

Requisiti

Cisco consiglia la conoscenza della **Cisco Software Defined Wide Area Network (SDWAN)** soluzione.

Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware. Questo documento è incentrato sulle piattaforme vEdge.

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.
Per i router Cisco Edge (router Cisco IOS® XE in modalità controller) , leggere .

Informazioni sul piano di controllo

Controllo proprietà locali controllo

Per controllare lo stato delle **Wide Area Network (WAN)** interfacce su un vEdge, usare il comando **show control local-properties wan-interface-list.**

In questo output, è possibile vedere la RFC 4787 **Network Address Translation (NAT) Type.**

Quando il vEdge è dietro un dispositivo NAT (firewall, router, ecc.), gli indirizzi IPv4 pubblici e privati, le porte di origine pubbliche e private vengono utilizzate per costruire i tunnel del piano dati**User Datagram Protocol (UDP)**.

È inoltre possibile verificare lo stato dell'interfaccia del tunnel, il colore e il numero massimo di connessioni di controllo configurate.

```
vEdge1# show control local-properties wan-interface-list

 NAT TYPE: E -- indicates End-point independent mapping
           A -- indicates Address-port dependent mapping
           N -- indicates Not learned
           Requires minimum two vbonds to learn the NAT type

           PUBLIC          PUBLIC PRIVATE         PRIVATE          PRIVATE                          MA
INTERFACE  IPv4            PORT   IPv4            IPv6             PORT    VS/VM COLOR        STATE CNT

-------------------------------------------------------------------------------------------------------
ge0/0      203.0.113.225   4501   10.19.145.2     ::               12386    1/1  gold           up    2
ge0/1      10.20.67.10     12426  10.20.67.10     ::               12426    0/0  mpls           up    2
```

Con questi dati, è possibile identificare alcune informazioni su come devono essere costruiti i tunnel di dati e sulle porte che ci si può aspettare (dalla prospettiva dei router) di utilizzare quando si formano i tunnel di dati.

Controlla connessioni di controllo

È importante assicurarsi che il colore che non forma i tunnel del piano dati abbia una connessione

di controllo stabilita con i controller nella sovrapposizione.

In caso contrario, vEdge non invia le **Transport Locator (TLOC)** informazioni a vSmart tramite **Overlay Management Protocol (OMP).**

È possibile verificare se funziona con l'uso del **show control connections** comando e cercare lo **connect** stato.

```
vEdge1# show control connections
                                                                                    PEER
PEER    PEER PEER            SITE       DOMAIN PEER                                 PRIV  PEER
TYPE    PROT SYSTEM IP       ID         ID     PRIVATE IP                           PORT  PUBLIC IP
-------------------------------------------------------------------------------------------------
vsmart  dtls 10.1.0.3         3          1      203.0.113.13                        12446 203.0.113.1
vbond   dtls -                0          0      203.0.113.12                        12346 203.0.113.1
vmanage dtls 10.1.0.1         1          0      203.0.113.14                        12646 203.0.113.1
```

Se l'interfaccia (che non forma i tunnel di dati) tenta di connettersi, risolverla con un avvio riuscito delle connessioni di controllo tramite quel colore.

In alternativa, impostare il comando **max-control-connections 0** nell'interfaccia selezionata sotto la sezione tunnel interface (interfaccia tunnel).

```
vpn 0
 interface ge0/1
  ip address 10.20.67.10/24
  tunnel-interface
   encapsulation ipsec
   color mpls restrict
   max-control-connections 0
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   no allow-service stun
  !
  no shutdown
 !
```

Nota: A volte è possibile utilizzare il **no control-connections** comando per raggiungere lo stesso obiettivo. Tuttavia, questo comando non stabilisce un numero massimo di connessioni di controllo. Questo comando è obsoleto rispetto alla versione 15.4 e non viene utilizzato con software più recente.

Overlay Management Protocol

Verificare che i TLOC OMP vengano annunciati dai bordi

Impossibile inviare i TLOC OMP perché l'interfaccia tenta di formare connessioni di controllo tramite tale colore e non è in grado di raggiungere i controller.

Verificare se il colore (che i dati utilizzano per il tunneling) invia il TLOC per quel particolare colore a vSmarts.

Utilizzare il comando **show omp tlocs advertised** per verificare i TLOC inviati ai peer OMP.

Esempio: Colori **mpls** e **gold**colori. Nessun TLOC inviato a vSmart per mpls a colori.

```
vEdge1# show omp tlocs advertised
C   -> chosen
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Stg -> staged
Inv -> invalid


ADDRESS                                                                    PSEUDO                    P
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP        P
-------------------------------------------------------------------------------------------------------
ipv4     10.1.0.5         gold             ipsec  0.0.0.0          C,Red,R   1       203.0.113.225    4
         10.1.0.2         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.20      1
         10.1.0.2         blue             ipsec  10.1.0.3          C,I,R     1       198.51.100.187   1
         10.1.0.30         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.30
         10.1.0.30         gold             ipsec  10.1.0.3          C,I,R     1       192.0.2.129
         10.1.0.4         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.40      1
         10.1.0.4         gold             ipsec  10.1.0.3          C,I,R     1       203.0.113.226    1
```

Esempio: Colori **mpls** e **gold**colori. Viene inviato TLOC per entrambi i colori.

```
vEdge2# show omp tlocs advertised
C   -> chosen
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Stg -> staged
Inv -> invalid
```

```
ADDRESS                                                                                  PSEUDO                  PU
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP          PO
-------------------------------------------------------------------------------------------------------------
ipv4     10.1.0.5         gold             ipsec  10.1.0.3          C,I,R     1       203.0.113.225     4
         10.1.0.2         mpls             ipsec  0.0.0.0         C,Red,R   1       10.20.67.20       12
         10.1.0.2         blue             ipsec  0.0.0.0         C,Red,R   1       198.51.100.187    12
         10.1.0.30         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.30
         10.1.0.30         gold             ipsec  10.1.0.3          C,I,R     1       192.0.2.129
         10.1.0.4         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.40       1
         10.1.0.4         gold             ipsec  10.1.0.3          C,I,R     1       203.0.113.226     1
```

Nota: Per qualsiasi informazione sul control plane generata localmente, il campo "FROM PEER" è impostato su 0.0.0.0. Quando si cercano informazioni originate localmente, assicurarsi che corrispondano in base a questo valore.

Verifica della ricezione e dell'annuncio vSmart dei TLOC

I TLOC vengono ora pubblicizzati sullo Smart Switch vSmart. Confermare di ricevere i TLOC dal peer corretto e di pubblicizzarli sull'altro vEdge.

Esempio: vSmart riceve i TLOC da 10.1.0.2 vEdge1.

<#root>

```
vSmart1# show omp tlocs received

C   -> chosen
I   -> installed

Red -> redistributed
Rej -> rejected
L   -> looped

R   -> resolved

S   -> stale
Ext -> extranet
Stg -> staged
Inv -> invalid


ADDRESS                                                                                  PSEUDO                  PU
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP          PO
-------------------------------------------------------------------------------------------------------------
ipv4     10.1.0.5         gold             ipsec  10.1.0.5          C,I,R     1       203.0.113.225     450

         10.1.0.2         mpls             ipsec  10.1.0.2          C,I,R     1       10.20.67.20       12


         10.1.0.2         blue             ipsec  10.1.0.2          C,I,R     1       198.51.100.187    12


         10.1.0.30         mpls             ipsec  10.1.0.30          C,I,R     1       10.20.67.30
         10.1.0.30         gold             ipsec  10.1.0.30          C,I,R     1       192.0.2.129
```

```
        10.1.0.4        mpls            ipsec  10.1.0.4        C,I,R     1       10.20.67.40      1
        10.1.0.4        gold            ipsec  10.1.0.4        C,I,R     1       203.0.113.226    1
```

Se i TLOC non vengono visualizzati o se vengono visualizzati altri codici, verificare quanto segue:

<#root>

```
vSmart-vIPtela-MEX# show omp tlocs received

C   -> chosen

I   -> installed

Red -> redistributed

Rej -> rejected

L   -> looped

R   -> resolved

S   -> stale
Ext -> extranet
Stg -> staged

Inv -> invalid


ADDRESS                                                                    PSEUDO                    P
FAMILY   TLOC IP          COLOR           ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP        P
-------------------------------------------------------------------------------------------------------
ipv4     10.1.0.5         gold            ipsec  10.1.0.5         C,I,R     1       203.0.113.225    4

         10.1.0.2         mpls            ipsec  10.1.0.2         C,I,R     1       10.20.67.20      12


         10.1.0.2         blue            ipsec  10.1.0.2         Rej,R,Inv 1       198.51.100.187   12


         10.1.0.30         mpls            ipsec  10.1.0.30         C,I,R     1       10.20.67.30
         10.1.0.30         gold            ipsec  10.1.0.30         C,I,R     1       192.0.2.129
         10.1.0.4         mpls            ipsec  10.1.0.4         C,I,R     1       10.20.67.40      1
         10.1.0.4         gold            ipsec  10.1.0.4         C,I,R     1       203.0.113.226    1
```

Verificare che non vi siano criteri che bloccano i TLOC.

show run policy control-policy - cercare eventuali elenchi di scelta che rifiutino i TLOC come **advertised** o **received** nella vSmart.

<#root>

```
vSmart1(config-policy)# sh config
policy
 lists
```

```
  tloc-list SITE20

   tloc 10.1.0.2 color blue encap ipsec

  !
 !
 control-policy SDWAN

  sequence 10
   match tloc
    tloc-list SITE20
   !
   action reject ---->

here we are rejecting the TLOC 10.1.0.2,blue,ipsec
   !
  !
  default-action accept
 !
apply-policy
 site-list SITE20

  control-policy SDWAN in ----->

the policy is applied to control traffic coming IN the vSmart, it will filter the tlocs before adding i
```

 Nota: Se un TLOC è **Rejected** o **Invalid**, non viene annunciato agli altri spigoli.

Assicurarsi che un criterio non filtri il TLOC quando viene annunciato da vSmart. È possibile notare che il TLOC viene ricevuto su vSmart, ma non viene visualizzato sull'altro vEdge.

Esempio 1: vSmart con TLOC in C,I,R.

<#root>

```
vSmart1# show omp tlocs

C   -> chosen
I   -> installed

Red -> redistributed
Rej -> rejected
L   -> looped

R   -> resolved

S   -> stale
Ext -> extranet
Stg -> staged
Inv -> invalid


ADDRESS                                                                PSEUDO                  P
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP        P
-------------------------------------------------------------------------------------------------------
ipv4     10.1.0.5         mpls             ipsec  10.1.0.5         C,I,R     1       10.20.67.10      1
         10.1.0.5         gold             ipsec  10.1.0.5         C,I,R     1       203.0.113.225    4
```

```
 10.1.0.2         mpls             ipsec  10.1.0.2         C,I,R     1       10.20.67.20      12386   10
         10.1.0.2         blue             ipsec  10.1.0.2         C,I,R     1       198.51.100.187   12


         10.1.0.30         mpls             ipsec  10.1.0.30         C,I,R     1       10.20.67.30
         10.1.0.30         gold             ipsec  10.1.0.30         C,I,R     1       192.0.2.129
         10.1.0.4         mpls             ipsec  10.1.0.4         C,I,R     1       10.20.67.40      1
         10.1.0.4         gold             ipsec  10.1.0.4         C,I,R     1       203.0.113.226    1
```

Esempio 2: vEdge1 non vede il TLOC di colore blu di vEdge2. Vede solo il TLOC MPLS.

<#root>

```
vEdge1# show omp tlocs
C   -> chosen
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Stg -> staged
Inv -> invalid

ADDRESS                                                                                  PSEUDO                  P
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP        P
-----------------------------------------------------------------------------------------------------------------
ipv4     10.1.0.5         mpls             ipsec  0.0.0.0          C,Red,R   1       10.20.67.10      1
         10.1.0.5         gold             ipsec  0.0.0.0          C,Red,R   1       203.0.113.225    4

         10.1.0.2         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.20      1


         10.1.0.30         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.30
         10.1.0.30         gold             ipsec  10.1.0.3          C,I,R     1       192.0.2.129
         10.1.0.4         mpls             ipsec  10.1.0.3          C,I,R     1       10.20.67.40      1
         10.1.0.4         gold             ipsec  10.1.0.3          C,I,R     1       203.0.113.226    1
```

Quando si controlla il criterio, è possibile verificare il motivo per cui il TLOC non viene visualizzato sul vEdge1.

<#root>

```
vSmart1# show running-config policy
policy
 lists
  tloc-list SITE20

   tloc 10.1.0.2 color blue encap ipsec

  !
  site-list SITE10
   site-id 10
  !
```

```
 !
 control-policy SDWAN
  sequence 10
   match tloc
    tloc-list SITE20
   !
   action reject
   !
  !
  default-action accept
 !
apply-policy
 site-list SITE10
  control-policy SDWAN out

 !
!
```

Rilevamento inoltro bidirezionale

Informazioni sul comando show bfd sessions

Di seguito sono riportati gli elementi chiave da cercare nell'output:

<#root>

```
vEdge-2# show bfd sessions
                                       SOURCE TLOC      REMOTE TLOC
SYSTEM IP        SITE ID  STATE       COLOR            COLOR            SOURCE IP
-----------------------------------------------------------------------------------------------------------

10.1.0.5         10       down        blue             gold             10.19.146.2


10.1.0.30         30       up          blue             gold             10.19.146.2
10.1.0.4         40       up          blue             gold             10.19.146.2
10.1.0.4         40       up          mpls             mpls             10.20.67.10
```

- **SYSTEM IP**: Peer system-ip
- **SOURCE and REMOTE TLOC COLOR**: Ciò è utile per sapere che cosa si prevede di ricevere e inviare nel TLOC.
- **SOURCE IP**: È l'IP di **private** origine. Se si è dietro un NAT, queste informazioni vengono visualizzate qui (possono essere visualizzate con l'uso di **show control local-properties** < **trong**="">).
  - **DST PUBLIC IP**: Si tratta della destinazione utilizzata da vEdge per formare il **Data Plane**

tunnel, indipendentemente dal fatto che si trovi dietro NAT o meno. (Esempio: Spigoli collegati direttamente a Internet o **Multi-Protocol Label Switching (MPLS)** collegamenti)

- **DST PUBLIC PORT** Porta pubblica NAT utilizzata dal server vEdge per formare il **Data Plane** tunnel verso il server vEdge remoto.
- **TRANSITIONS**: Numero di volte in cui lo stato della sessione BFD è stato modificato, da **NA** a **UP** e viceversa.

Comando show tunnel statistics

Il **show tunnel statistics** può visualizzare informazioni sui tunnel del piano dati. È possibile stabilire se inviare o ricevere pacchetti per un particolare tunnel IPSEC tra i bordi.

In questo modo è possibile capire se i pacchetti arrivano a ciascuna estremità e isolare i problemi di connettività tra i nodi.

Nell'esempio, quando si esegue il comando più volte, è possibile notare un incremento o nessun incremento nella **tx-pkts** barra o **rx-pkts**.

Suggerimento: Se il contatore per l'incremento di tx-pkts viene utilizzato, i dati vengono trasmessi al peer. Se il pkts rx non aumenta, significa che i dati non vengono ricevuti dal peer. In questo caso, controllare l'altra estremità e verificare se il tx-pkts viene incrementato.

<#root>

```
vEdge2# show tunnel statistics

TUNNEL                                  SOURCE  DEST                                                    T
PROTOCOL  SOURCE IP      DEST IP        PORT    PORT   SYSTEM IP    LOCAL COLOR      REMOTE COLOR     M
-------------------------------------------------------------------------------------------------------

ipsec     172.16.16.147  10.88.244.181  12386   12406  10.1.0.5     public-internet  default          1


ipsec     172.16.16.147  10.152.201.104 12386   63364  10.1.0.0  public-internet  default          1441
ipsec     172.16.16.147  10.152.204.31  12386   58851  10.1.0.7     public-internet  public-internet  1
ipsec     172.24.90.129  10.88.244.181  12426   12406  10.1.0.5     biz-internet     default          1
ipsec     172.24.90.129  10.152.201.104 12426   63364  10.1.0.0  biz-internet     default          1441
ipsec     172.24.90.129  10.152.204.31  12426   58851  10.1.0.7     biz-internet     public-internet  1

TUNNEL                                  SOURCE  DEST                                                    T
PROTOCOL  SOURCE IP      DEST IP        PORT    PORT   SYSTEM IP    LOCAL COLOR      REMOTE COLOR     M
-------------------------------------------------------------------------------------------------------

ipsec     172.16.16.147  10.88.244.181  12386   12406  10.1.0.5     public-internet  default          1


ipsec     172.16.16.147  10.152.201.104 12386   63364  10.1.0.0  public-internet  default          1441
ipsec     172.16.16.147  10.152.204.31  12386   58851  10.1.0.7     public-internet  public-internet  1
ipsec     172.24.90.129  10.88.244.181  12426   12406  10.1.0.5     biz-internet     default          1
ipsec     172.24.90.129  10.152.201.104 12426   63364  10.1.0.0  biz-internet     default          1441
ipsec     172.24.90.129  10.152.204.31  12426   58851  10.1.0.7     biz-internet     public-internet  1
```

Un altro comando utile è **show tunnel statistics bfd** quello di controllare il numero di pacchetti BFD inviati e ricevuti in un particolare tunnel data plane:

<#root>

```
vEdge1# show tunnel statistics bfd

                                                                                                  BFD   BFD
                                                        BFD       BFD                             PMTU  PMT
TUNNEL                                  SOURCE  DEST    ECHO TX   ECHO RX   BFD ECHO    BFD ECHO  TX    RX
PROTOCOL  SOURCE IP      DEST IP        PORT    PORT    PKTS      PKTS      TX OCTETS   RX OCTETS  PKTS  PKT
-------------------------------------------------------------------------------------------------------------
ipsec     192.168.109.4  192.168.109.5  4500    4500    0         0         0           0          0     0
ipsec     192.168.109.4  192.168.109.5  12346   12366   1112255   1112253   186302716   186302381  487   487
ipsec     192.168.109.4  192.168.109.7  12346   12346   1112254   1112252   186302552   186302210  487   487
ipsec     192.168.109.4  192.168.110.5  12346   12366   1112255   1112253   186302716   186302381  487   487
```

Elenco accessi

Un elenco degli accessi è un passaggio utile e necessario dopo aver esaminato l'**show bfd sessions**output.

Ora che gli IP e le porte pubblici e privati sono noti, è possibile creare una **Access Control List (ACL)** corrispondenza con SRC_PORT, DST_PORT, SRC_IP, DST_IP.

Ciò consente di verificare i messaggi BFD inviati e ricevuti.

Di seguito è riportato un esempio di configurazione di un ACL:

<#root>

```
policy
 access-list checkbfd-out
  sequence 10
   match
    source-ip         192.168.0.92/32
    destination-ip    198.51.100.187/32
    source-port       12426
    destination-port 12426
   !
   action accept
    count bfd-out-to-dc1-from-br1
   !
  !
  default-action accept
 !
 access-list checkbfd-in
  sequence 20
   match
    source-ip         198.51.100.187/32
    destination-ip    192.168.0.92/32
    source-port       12426
    destination-port 12426
```

```
   !
   action accept
    count bfd-in-from-dc1-to-br1
   !
  !
  default-action accept
 !
vpn 0
 interface ge0/0
   access-list checkbfd-in in
   access-list checkbfd-out out
  !
 !
!
```

Nell'esempio, questo ACL usa due sequenze. La sequenza 10 corrisponde ai messaggi BFD inviati da questo vEdge al peer. La sequenza 20 fa l'opposto.

e viene confrontata con le porte di origine (**Private**) e di destinazione (**Public**). Se vEdge utilizza NAT, verificare che le porte di origine e di destinazione siano corrette.

Per controllare i colpi su ogni contatore di sequenza, emettere il **show policy access-list counters**

<#root>

```
vEdge1# show policy access-list-counters

NAME      COUNTER NAME               PACKETS  BYTES
-----------------------------------------------------
checkbfd  bfd-out-to-dc1-from-br1    10       2048
          bfd-in-from-dc1-to-br1        0        0
```

Network Address Translation

Come utilizzare strumenti stun-client per rilevare mappe e filtri NAT.

Se sono stati completati tutti i passaggi e si è dietro NAT, il passaggio successivo consiste nell'identificare il **UDP NAT Traversal (RFC 4787) Map and Filter** comportamento.

Questo strumento viene utilizzato per individuare l'indirizzo IP esterno vEdge locale quando il vEdge si trova dietro un dispositivo NAT.

Questo comando ottiene un mapping delle porte per il dispositivo e, facoltativamente, individua le proprietà relative al NAT tra il dispositivo locale e un server (server pubblico: esempio di google stun server).

Nota: Per informazioni più dettagliate, visitare il sito:

<#root>

```
vEdge1# tools stun-client vpn 0 options "--mode full --localaddr 192.168.12.100 12386 --verbosity 2 stun
```

```
stunclient --mode full --localaddr 192.168.12.100 stun.l.google.com in VPN 0
Binding test: success
Local address: 192.168.12.100:12386
Mapped address: 203.0.113.225:4501
Behavior test: success
```

```
Nat behavior: Address Dependent Mapping
```

```
Filtering test: success
```

```
Nat filtering: Address and Port Dependent Filtering
```

Nelle versioni più recenti del software, la sintassi può essere leggermente diversa:

<#root>

```
vEdge1# tools stun-client vpn 0 options "--mode full --localaddr 192.168.12.100 --localport 12386 --verb
```

Nell'esempio, viene eseguito un test di rilevamento NAT completo con l'utilizzo della porta di origine UDP 12386 sul server Google STUN.

L'output di questo comando restituisce il comportamento NAT e il tipo di filtro NAT in base alla RFC 4787.

Nota: Quando si utilizza, **tools stun**tenere presente che, in caso contrario, il servizio STUN non funzionerà. Usare **allow-service stun** per lasciare passare i dati di stordimento.

<#root>

```
vEdge1# show running-config vpn 0 interface ge0/0
vpn 0
 interface ge0/0
  ip address 10.19.145.2/30
  !
  tunnel-interface
   encapsulation ipsec
   color gold
   max-control-connections 1
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   no allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
```

```
   allow-service stun

  !
  no shutdown
 !
!
```

Questo mostra la mappatura tra la terminologia STUN (Full-Cone NAT) e la RFC 4787 (NAT Behavioral for UDP).

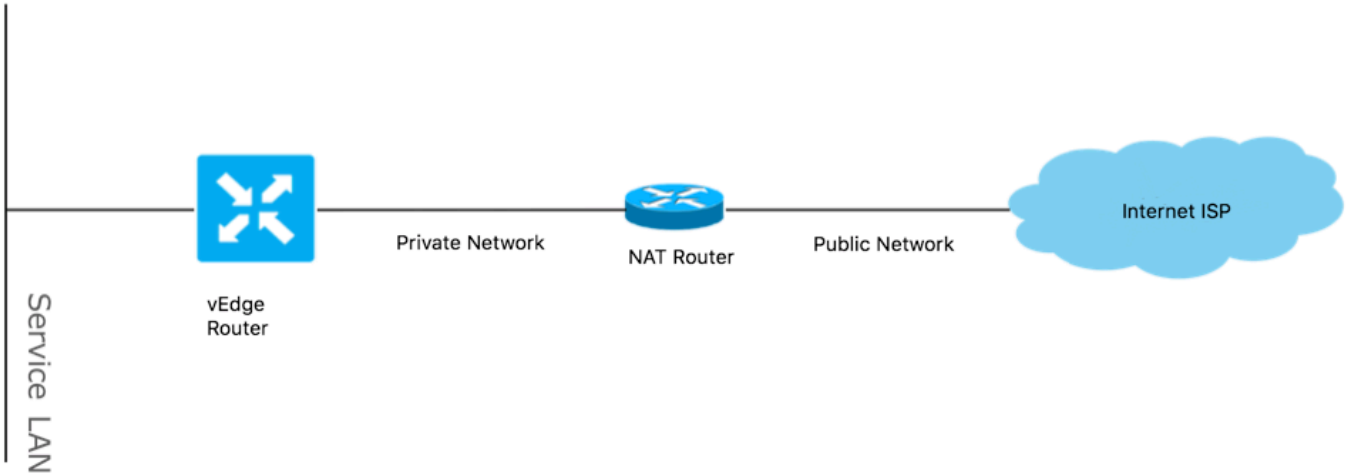
NAT Traversal Mapping Between used Viptela Terminologies		
STUN RFC 3489 Terminology	**RFC 4787 Terminology**	
	Mapping Behavior	**Filtering Behavior**
Full-cone NAT	Endpoint-Independent Mapping	Endpoint-Independent Filtering
Restricted Cone NAT	Endpoint-Independent Mapping	Address-Dependent Filtering
Port-Restricted Cone NAT	Endpoint-Independent Mapping	Address and Port-Dependent Filtering
Symmetric NAT	Address-and(or) Port-Dependent Mapping	Address-Dependent Filtering
		Address and Port-Dependent Filtering

Tipi NAT supportati per l'invio di tunnel Data Plane da CLI

Nella maggior parte dei casi, i colori pubblici come internet-biz o internet-pubblico possono essere collegati direttamente a internet.

In altri casi, è presente un dispositivo NAT dietro l'interfaccia WAN vEdge e l'effettivo Internet Service Provider.

In questo modo, vEdge può avere un IP privato e l'altro dispositivo (router, firewall, ecc.) può essere il dispositivo con gli indirizzi IP pubblici.



Se il tipo NAT non è corretto, potrebbe essere una delle cause più comuni che non consentono la formazione di tunnel Data Plane. Questi sono i tipi NAT supportati.

NAT Traversal Support		
Source	**Destination**	**Supported (YES/NO)**
Full-Cone NAT	Full-cone NAT	Yes
Full-Cone NAT	Restricted Cone NAT	Yes
Full-Cone NAT	Port-Restricted Cone NAT	Yes
Full-Cone NAT	Symmetric NAT	Yes
Restricted Cone NAT	Full-cone NAT	Yes
Restricted Cone NAT	Restricted Cone NAT	Yes
Restricted Cone NAT	Port-Restricted Cone NAT	Yes
Restricted Cone NAT	Symmetric NAT	Yes
Port-Restricted Cone NAT	Full-cone NAT	Yes
Port-Restricted Cone NAT	Restricted Cone NAT	Yes
Port-Restricted Cone NAT	Port-Restricted Cone NAT	Yes
Port-Restricted Cone NAT	Symmetric NAT	**No**
Symmetric NAT	Full-cone NAT	Yes
Symmetric NAT	Restricted Cone NAT	yes
Symmetric NAT	Port-Restricted Cone NAT	**No**
Symmetric NAT	Symmetric NAT	**No**

Firewall

Se il NAT è già stato controllato e non è incluso nei tipi di origine e destinazione non supportati, è possibile che un firewall blocchi le porte utilizzate per formare i **Data Plane** tunnel.

Verificare che queste porte siano aperte nel firewall per le connessioni Data Plane: **vEdge to vEdge Data Plane**:

UDP da 12346 a 13156

Per le connessioni di controllo da vEdge ai controller:

UDP da 12346 a 13156

da TCP 23456 a 24156

Accertarsi di aprire queste porte per completare correttamente la connessione dei tunnel del piano dati.

Quando si controllano le porte di origine e di destinazione utilizzate per i tunnel di Data Plane, è possibile utilizzare **show tunnel statistics** o **show bfd sessions | tab** ma non **show bfd sessions**utilizzare.

Non vengono visualizzate porte di origine, ma solo porte di destinazione come mostrato di seguito:

<#root>

```
vEdge1# show bfd sessions
                                       SOURCE TLOC      REMOTE TLOC
SYSTEM IP        SITE ID  STATE       COLOR            COLOR           SOURCE IP
-------------------------------------------------------------------------------------------------------
192.168.30.105   50       up          biz-internet     biz-internet    192.168.109.181
```

```
192.168.30.105   50       up          private1         private1         192.168.110.181

vEdge1# show bfd sessions | tab

                                        SRC    DST                    SITE
SRC IP           DST IP           PROTO  PORT   PORT   SYSTEM IP       ID    LOCAL COLOR   COLOR
-------------------------------------------------------------------------------------------------
192.168.109.181  192.168.109.182  ipsec  12346  12346  192.168.30.105  50    biz-internet  biz-internet
192.168.110.181  192.168.110.182  ipsec  12346  12346  192.168.30.105  50    private1      private1
```

Nota: Per ulteriori informazioni sulle porte firewall SD-WAN utilizzate, consultare qui.

Sicurezza

Se si nota che il contatore ACL aumenta sia in entrata che in uscita, controllare più iterazioni **show system statistics diff** and ensure there are no drops.

<#root>

```
vEdge1# show policy access-list-counters

NAME      COUNTER NAME               PACKETS  BYTES
-----------------------------------------------------

checkbfd  bfd-out-to-dc1-from-br1    55       9405



   bfd-in-from-dc1-to-br1     54      8478
```

In questo output, **rx_replay_integrity_drops** aumenta con ogni iterazione del **show system statistics diff** command.

<#root>

```
vEdge1#show system statistics diff

 rx_pkts : 5741427
 ip_fwd : 5952166
 ip_fwd_arp : 3
 ip_fwd_to_egress : 2965437
 ip_fwd_null_mcast_group : 26
 ip_fwd_null_nhop : 86846
 ip_fwd_to_cpu : 1413393
 ip_fwd_from_cpu_non_local : 15
 ip_fwd_rx_ipsec : 1586149
 ip_fwd_mcast_pkts : 26
 rx_bcast : 23957
 rx_mcast : 304
 rx_mcast_link_local : 240
 rx_implicit_acl_drops : 12832
 rx_ipsec_decap : 21
```

```
 rx_spi_ipsec_drops : 16


rx_replay_integrity_drops : 1586035


 port_disabled_rx : 2
 rx_invalid_qtags : 212700
 rx_non_ip_drops : 1038073
 pko_wred_drops : 3
 bfd_tx_record_changed : 23
 rx_arp_non_local_drops : 19893
 rx_arp_reqs : 294
 rx_arp_replies : 34330
 arp_add_fail : 263
 tx_pkts : 4565384
 tx_mcast : 34406
 port_disabled_tx : 3
 tx_ipsec_pkts : 1553753
 tx_ipsec_encap : 1553753
 tx_pre_ipsec_pkts : 1553753
 tx_pre_ipsec_encap : 1553753
 tx_arp_replies : 377
 tx_arp_reqs : 34337
 tx_arp_req_fail : 2
 bfd_tx_pkts : 1553675
 bfd_rx_pkts : 21
 bfd_tx_octets : 264373160
 bfd_rx_octets : 3600
 bfd_pmtu_tx_pkts : 78
 bfd_pmtu_tx_octets : 53052
 rx_icmp_echo_requests : 48
 rx_icmp_network_unreach : 75465
 rx_icmp_other_types : 47
 tx_icmp_echo_requests : 49655
 tx_icmp_echo_replies : 48
 tx_icmp_network_unreach : 86849
 tx_icmp_other_types : 7
vEdge1# show system statistics diff

 rx_pkts : 151
 ip_fwd : 157
 ip_fwd_to_egress : 75
 ip_fwd_null_nhop : 3
 ip_fwd_to_cpu : 43
 ip_fwd_rx_ipsec : 41
 rx_bcast : 1


rx_replay_integrity_drops : 41


 rx_invalid_qtags : 7
 rx_non_ip_drops : 21
 rx_arp_non_local_drops : 2
 tx_pkts : 114
 tx_ipsec_pkts : 40
 tx_ipsec_encap : 40
 tx_pre_ipsec_pkts : 40
 tx_pre_ipsec_encap : 40
 tx_arp_reqs : 1
 bfd_tx_pkts : 40
 bfd_tx_octets : 6800
```

```
 tx_icmp_echo_requests : 1
vEdge1# show system statistics diff

 rx_pkts : 126
 ip_fwd : 125
 ip_fwd_to_egress : 58
 ip_fwd_null_nhop : 3
 ip_fwd_to_cpu : 33
 ip_fwd_rx_ipsec : 36
 rx_bcast : 1
 rx_implicit_acl_drops : 1


rx_replay_integrity_drops : 35


 rx_invalid_qtags : 6
 rx_non_ip_drops : 22
 rx_arp_replies : 1
 tx_pkts : 97
 tx_mcast : 1
 tx_ipsec_pkts : 31
 tx_ipsec_encap : 31
 tx_pre_ipsec_pkts : 31
 tx_pre_ipsec_encap : 31
 bfd_tx_pkts : 32
 bfd_tx_octets : 5442
 rx_icmp_network_unreach : 3
 tx_icmp_echo_requests : 1
 tx_icmp_network_unreach : 3
vEdge1# show system statistics diff

 rx_pkts : 82
 ip_fwd : 89
 ip_fwd_to_egress : 45
 ip_fwd_null_nhop : 3
 ip_fwd_to_cpu : 24
 ip_fwd_rx_ipsec : 22
 rx_bcast : 1
 rx_implicit_acl_drops : 1


rx_replay_integrity_drops : 24


 rx_invalid_qtags : 2
 rx_non_ip_drops : 14
 rx_arp_replies : 1
 tx_pkts : 62
 tx_mcast : 1
 tx_ipsec_pkts : 24
 tx_ipsec_encap : 24
 tx_pre_ipsec_pkts : 24
 tx_pre_ipsec_encap : 24
 tx_arp_reqs : 1
 bfd_tx_pkts : 23
 bfd_tx_octets : 3908
 rx_icmp_network_unreach : 3
 tx_icmp_echo_requests : 1
 tx_icmp_network_unreach : 3
vEdge1# show system statistics diff

 rx_pkts : 80
```

```
 ip_fwd : 84
 ip_fwd_to_egress : 39
 ip_fwd_to_cpu : 20
 ip_fwd_rx_ipsec : 24


rx_replay_integrity_drops : 22


 rx_invalid_qtags : 3
 rx_non_ip_drops : 12
 tx_pkts : 66
 tx_ipsec_pkts : 21
 tx_ipsec_encap : 21
 tx_pre_ipsec_pkts : 21
 tx_pre_ipsec_encap : 21
 bfd_tx_pkts : 21
 bfd_tx_octets : 3571
```

Eseguire innanzitutto un'operazione **request security ipsec-rekey** su vEdge. Quindi, passare attraverso diverse iterazioni di **show system statistics diff** e vedere se si vede ancora **rx_replay_integrity_drops.**

In caso affermativo, verificare la configurazione di protezione.

<#root>

```
vEdge1# show running-config security
security
 ipsec
 authentication-type sha1-hmac ah-sha1-hmac
 !
!
```

Problemi dell'ISP con il traffico contrassegnato DSCP

Per impostazione predefinita, tutto il traffico di controllo e gestione dal router vEdge ai controller viene trasferito sulle connessioni DTLS o TLS e contrassegnato con un valore DSCP CS6 (48 decimali).

Per il traffico dei tunnel della postazione dati, i router vEdge utilizzano l'incapsulamento IPsec o GRE per scambiare il traffico dati.

Per il rilevamento degli errori del piano dati e la misurazione delle prestazioni, i router si inviano periodicamente pacchetti BFD.

Questi pacchetti BFD sono contrassegnati anche con un valore DSCP di CS6 (48 decimali).

Dal punto di vista dell'ISP, questo tipo di traffico è visto come traffico UDP con valore DSCP CS6, anche perché i router vEdge e i controller SD-WAN copiano il DSCP che contrassegna per

impostazione predefinita l'intestazione IP esterna.

Di seguito viene riportato l'aspetto che potrebbe assumere se tcpdump viene eseguito su un router ISP di transito:

<#root>

```
14:27:15.993766 IP (tos 0xc0, ttl 64, id 44063, offset 0, flags [DF], proto UDP (17), length 168)
    192.168.109.5.12366 > 192.168.20.2.12346: [udp sum ok] UDP, length 140
14:27:16.014900 IP (tos 0xc0, ttl 63, id 587, offset 0, flags [DF], proto UDP (17), length 139)
    192.168.20.2.12346 > 192.168.109.5.12366: [udp sum ok] UDP, length 111
14:27:16.534117 IP (tos 0xc0, ttl 63, id 0, offset 0, flags [DF], proto UDP (17), length 157)
    192.168.109.5.12366 > 192.168.110.6.12346: [no cksum] UDP, length 129
14:27:16.534289 IP (tos 0xc0, ttl 62, id 0, offset 0, flags [DF], proto UDP (17), length 150)
    192.168.110.6.12346 > 192.168.109.5.12366: [no cksum] UDP, length 122
```

Come si può vedere qui, tutti i pacchetti sono contrassegnati con il byte TOS 0xc0, noto anche come campo DS (che equivale al decimale 192, o 110 000 00 in binario).

I primi 6 bit di ordine superiore corrispondono ai bit DSCP (valore 48 in decimale o CS6).

I primi 2 pacchetti nell'output corrispondono a un tunnel del control plane e i 2 che rimangono, al traffico di un tunnel del data plane.

In base alla lunghezza del pacchetto e al marchio TOS, può concludere con grande sicurezza che si tratta di pacchetti BFD (direzioni RX e TX). Anche questi pacchetti sono contrassegnati con CS6.

Talvolta alcuni provider di servizi (in particolare i provider di servizi VPN MPLS L3/MPLS L2) mantengono SLA diversi e possono gestire in modo diverso classi di traffico basate su contrassegni DSCP.

Ad esempio, se si dispone di un servizio Premium per assegnare la priorità al traffico voce e di segnalazione DSCP EF e CS6.

Poiché il traffico prioritario viene quasi sempre monitorato, anche se la larghezza di banda totale di un uplink non viene superata, per questo tipo di traffico è possibile rilevare la perdita di pacchetti e quindi anche le sessioni BFD possono lampeggiare.

In alcuni casi, è stato rilevato che se la coda di priorità dedicata sul router del provider di servizi è ridotta, non si verificheranno cali del traffico normale (ad esempio, quando si esegue il comando ping semplice dal router vEdge).

Infatti, il traffico è contrassegnato dal valore DSCP predefinito 0, come mostrato di seguito (byte TOS):

<#root>

```
15:49:22.268044 IP (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto UDP (17), length 142)
```

```
    192.168.110.5.12366 > 192.168.109.7.12346: [no cksum] UDP, length 114
15:49:22.272919 IP (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto UDP (17), length 142)
    192.168.110.5.12366 > 192.168.109.7.12346: [no cksum] UDP, length 114
15:49:22.277660 IP (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto UDP (17), length 142)
    192.168.110.5.12366 > 192.168.109.7.12346: [no cksum] UDP, length 114
15:49:22.314821 IP (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto UDP (17), length 142)
    192.168.110.5.12366 > 192.168.109.7.12346: [no cksum] UDP, length 114
```

Ma allo stesso tempo, le sessioni del BFD si alternano:

<#root>

```
show bfd history

                                                    DST PUBLIC       DST PUBLIC
SYSTEM IP        SITE ID  COLOR            STATE     IP               PORT        ENCAP  TIME
-------------------------------------------------------------------------------------------------------
192.168.30.4     13       public-internet  up        192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  up        192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  down      192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  down      192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  up        192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  up        192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  down      192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  down      192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.4     13       public-internet  up        192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.6     13       public-internet  up        192.168.109.4    12346       ipsec  2019-05-01T03
192.168.30.6     13       public-internet  down      192.168.109.4    12346       ipsec  2019-05-01T03
```

E qui ping si rivela utile per risolvere i problemi:

<#root>

```
vedge2# tools nping vpn 0 options "--tos 0x0c --icmp --icmp-type echo --delay 200ms -c 100 -q" 192.168.1
Nping in VPN 0

Starting Nping 0.6.47 ( http://nmap.org/nping ) at 2019-05-07 15:58 CEST
Max rtt: 200.305ms | Min rtt: 0.024ms | Avg rtt: 151.524ms
Raw packets sent: 100 (2.800KB) | Rcvd: 99 (4.554KB) | Lost: 1 (1.00%)
Nping done: 1 IP address pinged in 19.83 seconds
```

Debug BFD

Se è necessaria un'analisi più approfondita, eseguire il debug di BFD sul router vEdge.

Forwarding Traffic Manager (FTM) è responsabile delle operazioni BFD sui router vEdge e quindi è necessario **debug ftm bfd**eseguire questa operazione.

Tutti gli output del debug sono memorizzati in un **/var/log/tmplog/vdebug** file e se si desidera che tali messaggi siano presenti sulla console (in modo simile al **terminal monitor** comportamento di Cisco IOS), è possibile utilizzare **monitor start /var/log/tmplog/vdebug.**

Per interrompere la registrazione, è possibile utilizzare **monitor stop /var/log/tmplog/vdebug**

Di seguito viene riportata la ricerca della sessione BFD nell'output a causa del timeout (TLOC remoto con indirizzo IP 192.168.110.6 non più raggiungibile):

<#root>

```
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1008]: BFD-session TNL 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_proc_tunnel_public_tloc_msg[252]: tun_rec_index
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_increment_wanif_bfd_flap[2427]: BFD-session TNL
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1119]: BFD-session TNL 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1140]: Attempting to add TLOC : from_tt
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32771:
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32770:
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session c
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32771:3277
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session c
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32770:3277
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32771:32772) src 192.168.11
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32770:32772) src 192.168.10
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1008]: BFD-session TNL 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_proc_tunnel_public_tloc_msg[252]: tun_rec_index
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_increment_wanif_bfd_flap[2427]: BFD-session TNL
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1119]: BFD-session TNL 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1140]: Attempting to add TLOC : from_tt
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32771:
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32770:
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session c
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32771:3277
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session c
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32770:3277
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32771:32772) src 192.168.11
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32770:32772) src 192.168.10
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_send_bfd_msg[499]: Sending BFD notification Down
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1140]: Attempting to add TLOC : from_tt
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32771:
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32770:
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1285]: UPDATE local tloc
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session c
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32771:3277
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session c
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32770:3277
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32771:32772) src 192.168.11
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32770:32772) src 192.168.10
log:local7.info: May  7 16:23:09 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002: Notification: 5A
log:local7.info: May  7 16:23:09 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002: Notification: 5A
```

Un altro valido strumento di debug per abilitare il debug degli **Tunnel Traffic Manager (TTM)** eventi è **debug ttm events**il.

Ecco come appare l'**BFD DOWN**evento dal punto di vista di TTM:

<#root>

```
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM Msg LINK_E
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[413]:     Remote-TLOC: 192.16
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM Msg LINK_E
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[413]:     Remote-TLOC: 192.16
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM Msg BFD, C
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[402]:     TLOC: 192.168.30.6
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_af_tloc_db_bfd_status[234]:   BFD message: I SAY
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Sent TTM Msg TLOC_ADD,
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[213]:     TLOC: 192.168.30.6
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[217]:       Attributes: GROUP
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[220]:         Preference: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[223]:         Weight: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[226]:         Gen-ID: 2147483
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[229]:         Version: 2
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[232]:         Site-ID: 13
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[235]:         Carrier: 4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[241]:         Restrict: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[249]:         Group: Count: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[262]:         Groups: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[269]:         TLOCv4-Public:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[273]:         TLOCv4-Private:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[277]:         TLOCv6-Public:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[281]:         TLOCv6-Private:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[285]:         TLOC-Encap: ips
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[295]:           Authenticatio
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[312]:           SPI 334, Flag
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[317]:           Number of pro
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[328]:           Number of enc
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:           Encrypt type|
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:           Encrypt type|
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[339]:           Number of int
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[344]:           integrity typ
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[349]:       #Paths: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Sent TTM Msg TLOC_ADD,
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[213]:     TLOC: 192.168.30.6
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[217]:       Attributes: GROUP
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[220]:         Preference: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[223]:         Weight: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[226]:         Gen-ID: 2147483
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[229]:         Version: 2
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[232]:         Site-ID: 13
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[235]:         Carrier: 4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[241]:         Restrict: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[249]:         Group: Count: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[262]:         Groups: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[269]:         TLOCv4-Public:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[273]:         TLOCv4-Private:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[277]:         TLOCv6-Public:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[281]:         TLOCv6-Private:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[285]:         TLOC-Encap: ips
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[295]:           Authenticatio
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[312]:           SPI 334, Flag
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[317]:           Number of pro
```

```
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[328]:           Number of enc
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:           Encrypt type|
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:           Encrypt type|
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[339]:           Number of int
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[344]:           integrity typ
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[349]:       #Paths: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Sent TTM Msg TLOC_ADD,
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[213]:     TLOC: 192.168.30.6
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[217]:       Attributes: GROUP
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[220]:         Preference: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[223]:         Weight: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[226]:         Gen-ID: 2147483
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[229]:         Version: 2
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[232]:         Site-ID: 13
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[235]:         Carrier: 4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[241]:         Restrict: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[249]:         Group: Count: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[262]:         Groups: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[269]:         TLOCv4-Public:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[273]:         TLOCv4-Private:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[277]:         TLOCv6-Public:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[281]:         TLOCv6-Private:
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[285]:         TLOC-Encap: ips
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[295]:           Authenticatio
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[312]:           SPI 334, Flag
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[317]:           Number of pro
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[328]:           Number of enc
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:           Encrypt type|
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:           Encrypt type|
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[339]:           Number of int
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[344]:           integrity typ
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[349]:       #Paths: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Sent TTM Msg DATA_DEVIC
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[431]:     Device: 192.168.30.
log:local7.info: May  7 16:58:19 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002: Notification: 5/
log:local7.info: May  7 16:58:20 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002: Notification: 5/
```

Utilizzare Packet-Trace per acquisire pacchetti BFD (versione 20.5 e successive)

Un altro utile strumento introdotto nella versione 20.5.1 e successive del software è packet-trace per vEdges.
Poiché la sessione BFD utilizza le stesse porte standard, generalmente 12346, è più semplice filtrare in base all'indirizzo IP del peer.

Ad esempio:

<#root>

```
vedge# show bfd sessions
                                       SOURCE TLOC      REMOTE TLOC
SYSTEM IP        SITE ID  STATE       COLOR            COLOR            SOURCE IP
-------------------------------------------------------------------------------------------------------
```

```
10.4.4.1         101      up          default          default          192.168.16.29
10.4.4.2         102      up          default          default          192.168.16.29
```

Packet-trace verrà configurato:

vedge# debug packet-trace condition ingress-if ge0/0 vpn 0 source-ip 192.168.29.39
avvio condizione vedge# debug packet-trace
arresto condizione di traccia dei pacchetti di debug di vedge#

I risultati possono essere visualizzati utilizzando i comandi show indicati di seguito. Per i pacchetti in entrata, è presente un flag 'isBFD' impostato su '1' (true) per il traffico BFD.

<#root>

```
vedge# show packet-trace statistics
packet-trace statistics 0
 source-ip             192.168.29.39
 source-port           12346
 destination-ip        192.168.16.29
 destination-port      12346
 source-interface      ge0_0
 destination-interface loop0.1
 decision              FORWARD
 duration              25
packet-trace statistics 1
 source-ip             192.168.29.39
 source-port           12346
 destination-ip        192.168.16.29
 destination-port      12346
 source-interface      ge0_0
 destination-interface loop0.1
 decision              FORWARD
 duration              14
packet-trace statistics 2
 source-ip             192.168.29.39
 source-port           12346
 destination-ip        192.168.16.29
 destination-port      12346
 source-interface      ge0_0
 destination-interface loop0.1
 decision              FORWARD
 duration              14

vedge# show packet-trace detail 0
================================================================================================================
Pkt-id           src_ip(ingress_if)            dest_ip(egress_if)           Duration           Decision
================================================================================================================
0         192.168.29.39:12346 (ge0_0)    192.168.16.29:12346 (loop0.1)        25 us             FORWARD
INGRESS_PKT:
00 50 56 84 79 be 00 50 56 84 3c b5 08 00 45 c0 00 96 ab 40 40 00 3f 11 e0 c1 c0 a8 1d 27 c0
a8 10 1d 30 3a 30 3a 00 82 00 00 a0 00 01 02 00 00 0e 3f 4b 65 07 bc 61 03 38 71 93 53 58
88 d8 08 41 95 7c 1a ff 8b cc b4 d0 d8 61 44 40 67 cc 1a 01 fd 1f c4 45 95 ea 7e 15 c9 08
2e b6 63 84 00
EGRESS_PKT:
a1 5e fe 11 00 00 00 00 00 00 00 00 00 04 00 0c 04 00 41 01 02 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 04 00 00 00 00 00 00 00 02 00 3a 30 3a 30 1d 10 a8 c0 00 00 00 00 00 00
00 00 00 00 00 00 01 00 00 00 27 1d a8 c0 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00
```

```
a4 00 01 00 00
Feature Data
-----------------------------------
TOUCH : fp_proc_packet
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_proc_packet2
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_ip_forward
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_ipsec_decrypt
core_id: 2
DSCP: 48
-----------------------------------
FP_TRACE_FEAT_IPSEC_DATA:
src_ip : 192.168.29.39
src_port : 3784
dst_ip : 192.168.16.29
dst_port : 3784
isBFD : 1
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_send_pkt
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_hw_x86_pkt_free
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_proc_remote_bfd_
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : BFD_ECHO_REPLY
core_id: 2
DSCP: 48
-----------------------------------
TOUCH : fp_hw_x86_pkt_free
core_id: 2
DSCP: 48
```

I pacchetti BFD in uscita vengono acquisiti in modo simile. Questi risultati identificano il tipo specifico, una richiesta echo o una risposta.

vedge# debug packet-trace condizione vpn 0 destination-ip 192.168.29.39
avvio condizione vedge# debug packet-trace
arresto condizione di traccia dei pacchetti di debug di vedge#

<#root>

```
vedge# show packet-trace statistics
packet-trace statistics 0
 source-ip             192.168.16.29
 source-port           3784
 destination-ip        192.168.29.39
 destination-port      3784
 source-interface      loop0.0
 destination-interface ge0_0
 decision              FORWARD
 duration              15
packet-trace statistics 1
 source-ip             192.168.16.29
 source-port           3784
 destination-ip        192.168.29.39
 destination-port      3784
 source-interface      loop0.0
 destination-interface ge0_0
 decision              FORWARD
 duration              66
packet-trace statistics 2
 source-ip             192.168.16.29
 source-port           3784
 destination-ip        192.168.29.39
 destination-port      3784
 source-interface      loop0.0
 destination-interface ge0_0
 decision              FORWARD
 duration              17

vedge# show packet-trace details 0
================================================================================================================
Pkt-id           src_ip(ingress_if)            dest_ip(egress_if)           Duration           Decision
================================================================================================================
0         192.168.16.29:3784 (loop0.0)   192.168.29.39:3784 (ge0_0)           15 us              FORWARD
INGRESS_PKT:
45 c0 00 4f 00 00 40 00 ff 11 cc 48 c0 a8 10 1d c0 a8 1d 27 0e c8 0e c8 00 3b 00 00 80 c0 07
00 00 00 00 01 00 00 00 01 00 0f 42 40 00 0f 42 40 00 0f 42 40 01 00 0c 01 00 00 1d 3b b1
c9 89 d7 03 00 0f c0 a8 10 1d 30 3a c0 a8 1d 27 30 3a a3 96 07 3b 47 1c 60 d1 d5 76 4c 72
78 1f 9a 0d 00
EGRESS_PKT:
00 50 56 84 3c b5 00 50 56 84 79 be 08 00 45 c0 00 96 ab 40 40 00 3f 11 e0 c1 c0 a8 10 1d c0
a8 1d 27 30 3a 30 3a 00 82 00 00 a0 00 01 01 00 00 5c 3d 88 9a c7 28 23 1b e6 18 ea fe 73
1b b9 e3 79 bf d9 f4 72 41 96 c1 47 07 44 56 77 5a a2 fb 43 59 c1 97 59 47 62 21 77 d4 f4
47 8b 30 b0 00
Feature Data
-----------------------------------
TOUCH : fp_send_bfd_pkt
core_id: 0
DSCP: 48
-----------------------------------
TOUCH : BFD_ECHO_REPLY
core_id: 0
DSCP: 48
-----------------------------------
TOUCH : fp_ipsec_loopback_f
core_id: 0
DSCP: 48
-----------------------------------
TOUCH : fp_send_pkt
core_id: 0
DSCP: 48
-----------------------------------
```

```
TOUCH : fp_ip_forward
core_id: 2
DSCP: 48
----------------------------------
TOUCH : fp_send_ip_packet
core_id: 2
DSCP: 48
----------------------------------
TOUCH : fp_send_pkt
core_id: 2
DSCP: 48
----------------------------------
TOUCH : fp_hw_x86_pkt_free
core_id: 2
DSCP: 48
vedge# show packet-trace details 1
============================================================================================================
Pkt-id           src_ip(ingress_if)            dest_ip(egress_if)           Duration           Decision
============================================================================================================
1         192.168.16.29:3784 (loop0.0)   192.168.29.39:3784 (ge0_0)           66 us              FORWARD
INGRESS_PKT:
45 c0 00 56 00 00 40 00 ff 11 cc 41 c0 a8 10 1d c0 a8 1d 27 0e c8 0e c8 00 42 00 00 80 c0 07
00 00 00 00 01 00 00 00 01 00 0f 42 40 00 0f 42 40 00 0f 42 40 01 00 0c 00 00 00 1d b8 35
a8 09 88 03 00 0f c0 a8 10 1d 30 3a c0 a8 1d 27 30 3a 04 00 07 01 00 05 a6 38 ff 7e 06 1e
da 23 19 d5 00
EGRESS_PKT:
00 50 56 84 3c b5 00 50 56 84 79 be 08 00 45 c0 00 9d ab 40 40 00 3f 11 e0 ba c0 a8 10 1d c0
a8 1d 27 30 3a 30 3a 00 89 00 00 a0 00 01 01 00 00 5c 3e 2d 3b 9e 81 aa 10 26 54 7f 47 5c
d8 81 4f 23 2e 3c 39 1e 94 b2 f4 fb a4 ba 98 54 73 99 8f 2e 95 d7 69 fb 91 41 96 93 03 5b
a4 e4 e8 82 00
Feature Data
----------------------------------
TOUCH : fp_send_bfd_pkt
core_id: 0
DSCP: 48
----------------------------------
TOUCH : BFD_ECHO_REQUEST
core_id: 0
DSCP: 48
----------------------------------
TOUCH : fp_ipsec_loopback_f
core_id: 0
DSCP: 48
----------------------------------
TOUCH : fp_send_pkt
core_id: 0
DSCP: 48
----------------------------------
TOUCH : fp_ip_forward
core_id: 2
DSCP: 48
----------------------------------
TOUCH : fp_send_ip_packet
core_id: 2
DSCP: 48
----------------------------------
TOUCH : fp_send_pkt
core_id: 2
DSCP: 48
----------------------------------
TOUCH : fp_hw_x86_pkt_free
core_id: 2
```

DSCP: 48

Informazioni correlate

- Documentazione e supporto tecnico – Cisco Systems

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).