Fehlerbehebung bei Problemen mit der bidirektionalen Weiterleitungserkennung und Datenebenenverbindungen

Inhalt

Einführung

In diesem Dokument werden Verbindungsprobleme auf Datenebene beschrieben, die auf vEdge-Routern auftreten können, nachdem Sie erfolgreich eine Verbindung zur Steuerungsebene hergestellt haben. Es besteht jedoch weiterhin keine Verbindung auf Datenebene zwischen den Standorten.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse der SDWAN-Lösung (Software Defined Wide Area Network) von Cisco zu verfügen.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardwareversionen beschränkt.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

> **Hinweis**: Alle in diesem Dokument aufgeführten Befehlsausgaben stammen von vEdge-Routern. Der Ansatz zur Fehlerbehebung ist jedoch für den Router identisch, auf dem die IOS®-XE SDWAN-Software ausgeführt wird. Verwenden Sie das **sdwan**-Schlüsselwort, um die gleichen Ausgaben für die IOS®-XE SDWAN-Software zu erhalten. Beispiel: **Anzeigen von SDWAN-Steuerungsverbindungen** anstelle von **Steuerelementverbindungen**.

Informationen zur Kontrollebene

Lokale Steuerelementeigenschaften überprüfen

Um den Status der WAN-Schnittstellen (Wide Area Network) auf einem vEdge zu überprüfen, verwenden Sie den Befehl **show control local-properties wan-interface-list**. In dieser Ausgabe wird der RFC 4787 Network Address Translation (NAT)-Typ angezeigt. Wenn sich der vEdge hinter einem NAT-Gerät (Firewall, Router usw.) befindet, werden für den Aufbau der Datenebenentunnels Public und Private IPv4-Adressen sowie UDP-Ports (Public und Private Source User Datagram Protocol) verwendet. Sie können auch den Zustand der Tunnelschnittstelle, die Farbe und die maximale Anzahl der konfigurierten Steuerverbindungen finden.

```
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
           Note: Requires minimum two vbonds to learn the NAT type

           PUBLIC          PUBLIC PRIVATE         PRIVATE          PRIVATE
MAX   RESTRICT/           LAST        SPI TIME   NAT  VM
INTERFACE  IPv4            PORT   IPv4            IPv6             PORT    VS/VM COLOR
STATE CNTRL  CONTROL/   LR/LB   CONNECTION   REMAINING  TYPE CON

STUN                                          PRF
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------
ge0/0      203.0.113.225   4501   10.19.145.2     ::               12386    1/1  gold
up    2    no/yes/no   No/No  7:02:55:13   0:09:02:29  N    5
ge0/1      10.20.67.10     12426  10.20.67.10     ::               12426    0/0  mpls
up    2    yes/yes/no  No/No  0:00:00:01   0:11:40:16  N    5
```

Anhand dieser Daten können Sie bestimmte Informationen darüber identifizieren, wie die Datentunnel erstellt werden müssen und welche Ports aus Routerperspektive für die Verwendung bei der Erstellung der Datentunnel benötigt werden.

Steuerungsverbindungen überprüfen

Es ist wichtig sicherzustellen, dass die Farbe, die keine Datenebenentunnel bildet, über eine Steuerungsverbindung mit den Controllern im Overlay verfügt. Andernfalls sendet der vEdge keine TLOC-Informationen (Transport Locator) über das Overlay Management Protocol (OMP) an den vSmart. Sie können mithilfe des Befehls **show control connections** sicherstellen, ob die Verbindung aktiv ist, und nach der **Verbindung** mit dem Status suchen.

```
vEdge1# show control connections
                                                                                PEER
PEER                                          CONTROLLER
PEER    PEER PEER            SITE       DOMAIN PEER                             PRIV
PEER                                      PUB                                        GROUP
TYPE    PROT SYSTEM IP       ID         ID     PRIVATE IP                       PORT
PUBLIC IP                                 PORT  LOCAL COLOR     STATE           UPTIME      ID
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
--
vsmart  dtls 1.1.1.3         3          1      203.0.113.13                          12446
203.0.113.13                           12446 gold            up              7:03:18:31  0
vbond   dtls -               0          0      203.0.113.12                          12346
203.0.113.12                           12346 mpls            connect                    0
vmanage dtls 1.1.1.1         1          0      203.0.113.14                          12646
203.0.113.14                           12646 gold            up              7:03:18:31  0
```

Wenn die Schnittstelle, die keine Datentunnel bildet, versucht, eine Verbindung herzustellen, können Sie sie lösen, indem Sie die Steuerungsverbindungen über diese Farbe erfolgreich aufrufen. Alternativ können Sie die **max-control-connections 0** in der ausgewählten Schnittstelle im Tunnelschnittstellenabschnitt umgehen.

```
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
 !
```

> **Hinweis**: Manchmal können Sie den Befehl **no control-connections** verwenden, um dasselbe Ziel zu erreichen. Mit diesem Befehl wird jedoch keine maximale Anzahl von Steuerungsverbindungen festgelegt. Dieser Befehl ist seit 15.4 veraltet und sollte nicht auf neuerer Software verwendet werden.

Overlay Management Protocol

Überprüfen Sie, ob die OMP-TLOCs über die vEdges angezeigt werden.

Wie Sie im vorherigen Schritt bemerkt haben, können OMP TLOCs nicht gesendet werden, da die Schnittstelle versucht, über diese Farbe Steuerungsverbindungen herzustellen, und nicht in der Lage ist, die Controller zu erreichen. Prüfen Sie also, ob die Farbe, in der die Datentunnel nicht funktionieren, oder ob sie auftaucht, den TLOC für diese bestimmte Farbe an vSmarts sendet. Verwenden Sie den Befehl **show omp tlocs angekündigte** , um die TLOCs zu überprüfen, die an die OMP-Peers gesendet werden.

Beispiel: Farben **MPLS** und **Gold**. Für Farbkombinationen wird kein TLOC an vSmart gesendet.

```
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


PUBLIC           PRIVATE
ADDRESS                                                                                PSEUDO
PUBLIC                    PRIVATE  PUBLIC  IPV6    PRIVATE  IPV6     BFD
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP
PORT    PRIVATE IP       PORT     IPV6    PORT    IPV6     PORT     STATUS
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
ipv4     1.1.1.10         gold             ipsec  0.0.0.0          C,Red,R   1
203.0.113.225    4501    10.19.145.2      12386    ::      0       ::       0        up
         1.1.1.20         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.20
12386   10.20.67.20      12386    ::      0       ::       0        down
         1.1.1.20         blue             ipsec  1.1.1.3          C,I,R     1
198.51.100.187   12406   10.19.146.2      12406    ::      0       ::       0        up
         1.1.1.30         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.30
12346   10.20.67.30      12346    ::      0       ::       0        down
         1.1.1.30         gold             ipsec  1.1.1.3          C,I,R     1       192.0.2.129
12386   192.0.2.129      12386    ::      0       ::       0        up
         1.1.1.40         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.40
12426   10.20.67.40      12426    ::      0       ::       0        down
         1.1.1.40         gold             ipsec  1.1.1.3          C,I,R     1
203.0.113.226    12386   203.0.113.226    12386    ::      0       ::       0        up
```

Beispiel: Farben **MPLS** und **Gold**. TLOC wird für beide Farben gesendet.

```
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
```

```
PUBLIC           PRIVATE
ADDRESS                                                                    PSEUDO
PUBLIC                   PRIVATE  PUBLIC  IPV6    PRIVATE  IPV6     BFD
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP
PORT    PRIVATE IP       PORT     IPV6    PORT    IPV6     PORT     STATUS
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
ipv4     1.1.1.10         gold             ipsec  1.1.1.3          C,I,R     1
203.0.113.225    4501    10.19.145.2      12386    ::      0       ::       0        up
         1.1.1.20         mpls             ipsec  0.0.0.0          C,Red,R   1       10.20.67.20
12386   10.20.67.20      12386    ::      0       ::       0        up
         1.1.1.20         blue             ipsec  0.0.0.0          C,Red,R   1
198.51.100.187   12406   10.19.146.2      12406    ::      0       ::       0        up
         1.1.1.30         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.30
12346   10.20.67.30      12346    ::      0       ::       0        up
         1.1.1.30         gold             ipsec  1.1.1.3          C,I,R     1       192.0.2.129
     12386   192.0.2.129      12386    ::      0       ::       0        up
         1.1.1.40         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.40
12426   10.20.67.40      12426    ::      0       ::       0        up
         1.1.1.40         gold             ipsec  1.1.1.3          C,I,R     1
203.0.113.226    12386   203.0.113.226    12386    ::      0       ::       0        up
```

Hinweis: Für alle lokal generierten Kontrollebeneninformationen wird das Feld "FROM PEER" (Von PEER) auf 0.0.0.0 festgelegt. Wenn Sie nach lokal erstellten Informationen suchen, stellen Sie sicher, dass diese auf Grundlage dieses Werts übereinstimmen.

Überprüfen Sie, ob vSmart die TLOCs empfängt und ankündigt

Nachdem Sie wissen, dass Ihre TLOCs an den vSmart weitergeleitet werden, bestätigen Sie, dass der vSmart TLOCs vom richtigen Peer empfängt und dem anderen vEdge ankündigt.

Beispiel: vSmart empfängt die TLOCs vom 1.1.1.20 vEdge1.

```
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
```

```
PUBLIC           PRIVATE
ADDRESS                                                                    PSEUDO
PUBLIC                   PRIVATE  PUBLIC  IPV6    PRIVATE  IPV6     BFD
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP
PORT    PRIVATE IP       PORT     IPV6    PORT    IPV6     PORT     STATUS
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
ipv4     1.1.1.10         gold             ipsec  1.1.1.10         C,I,R     1
203.0.113.225  4501    10.19.145.2      12386    ::      0       ::       0        -
         1.1.1.20         mpls             ipsec  1.1.1.20         C,I,R     1       10.20.67.20
12386   10.20.67.20      12386    ::      0       ::       0        -
         1.1.1.20         blue             ipsec  1.1.1.20         C,I,R     1
198.51.100.187   12406   10.19.146.2      12406    ::      0       ::       0        -
```

```
        1.1.1.30         mpls            ipsec  1.1.1.30        C,I,R     1       10.20.67.30
12346   10.20.67.30      12346    ::      0       ::       0        -
        1.1.1.30         gold            ipsec  1.1.1.30        C,I,R     1       192.0.2.129
12386   192.0.2.129      12386    ::      0       ::       0        -
        1.1.1.40         mpls            ipsec  1.1.1.40        C,I,R     1       10.20.67.40
12426   10.20.67.40      12426    ::      0       ::       0        -
        1.1.1.40         gold            ipsec  1.1.1.40        C,I,R     1
203.0.113.226    12386   203.0.113.226    12386    ::      0       ::       0        -
```

Falls Sie die TLOCs nicht sehen oder hier andere Codes sehen, können Sie diese überprüfen:

```
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


PUBLIC           PRIVATE
ADDRESS                                                                          PSEUDO
PUBLIC                   PRIVATE  PUBLIC  IPV6    PRIVATE  IPV6     BFD
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP
PORT    PRIVATE IP       PORT     IPV6    PORT    IPV6     PORT     STATUS
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
ipv4     1.1.1.10         gold            ipsec  1.1.1.10        C,I,R     1
203.0.113.225    4501    10.19.145.2      12386    ::      0       ::       0        -
         1.1.1.20         mpls            ipsec  1.1.1.20        C,I,R     1       10.20.67.20
12386   10.20.67.20      12386    ::      0       ::       0        -
         1.1.1.20         blue            ipsec  1.1.1.20        Rej,R,Inv 1
198.51.100.187   12406   10.19.146.2      12406    ::      0       ::       0        -
         1.1.1.30         mpls            ipsec  1.1.1.30        C,I,R     1       10.20.67.30
12346   10.20.67.30      12346    ::      0       ::       0        -
         1.1.1.30         gold            ipsec  1.1.1.30        C,I,R     1       192.0.2.129
    12386   192.0.2.129      12386    ::      0       ::       0        -
         1.1.1.40         mpls            ipsec  1.1.1.40        C,I,R     1       10.20.67.40
12426   10.20.67.40      12426    ::      0       ::       0        -
         1.1.1.40         gold            ipsec  1.1.1.40        C,I,R     1
203.0.113.226    12386   203.0.113.226    12386    ::      0       ::       0        -
```

Überprüfen Sie, ob es keine Richtlinie gibt, die die TLOCs blockiert.

show run policy control-policy-look for any tloc list, which weigert Ihre TLOCs, im vSmart angekündigt oder empfangen zu werden.

```
vSmart1(config-policy)# sh config
policy
 lists
  tloc-list SITE20
   tloc 1.1.1.20 color blue encap ipsec
  !
 !
 control-policy SDWAN
```

```
  sequence 10
   match tloc
    tloc-list SITE20
   !
   action reject ----> here we are rejecting the TLOC 1.1.1.20,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 it to the OMP table.
```

Hinweis: Wenn ein TLOC abgelehnt oder ungültig wird, wird er nicht an die anderen vEdges weitergegeben.

Stellen Sie sicher, dass eine Richtlinie den TLOC nicht filtert, wenn er vom vSmart angekündigt wird. Sie können sehen, dass die TLOC im vSmart empfangen wird, im anderen vEdge jedoch nicht.

Beispiel 1: vSmart mit TLOC in C,I,R

```
vSmart1# show omp tlocs
C   -> chosen
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Stg -> staged
Inv -> invalid


PUBLIC           PRIVATE
ADDRESS                                                                                PSEUDO
PUBLIC                    PRIVATE  PUBLIC  IPV6    PRIVATE  IPV6     BFD
FAMILY   TLOC IP          COLOR             ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP
PORT    PRIVATE IP       PORT     IPV6    PORT    IPV6     PORT     STATUS
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
ipv4     1.1.1.10         mpls              ipsec  1.1.1.10         C,I,R     1       10.20.67.10
12406   10.20.67.10      12406    ::      0       ::       0         -
         1.1.1.10         gold              ipsec  1.1.1.10         C,I,R     1
203.0.113.225    4501    10.19.145.2      12386    ::      0       ::       0        -
         1.1.1.20         mpls              ipsec  1.1.1.20         C,I,R     1       10.20.67.20
12386   10.20.67.20      12386    ::      0       ::       0         -
         1.1.1.20         blue              ipsec  1.1.1.20         C,I,R     1
198.51.100.187   12426   10.19.146.2      12426    ::      0       ::       0        -
         1.1.1.30         mpls              ipsec  1.1.1.30         C,I,R     1       10.20.67.30
12346   10.20.67.30      12346    ::      0       ::       0         -
         1.1.1.30         gold              ipsec  1.1.1.30         C,I,R     1       192.0.2.129
12386   192.0.2.129      12386    ::      0       ::       0         -
         1.1.1.40         mpls              ipsec  1.1.1.40         C,I,R     1       10.20.67.40
12426   10.20.67.40      12426    ::      0       ::       0         -
         1.1.1.40         gold              ipsec  1.1.1.40         C,I,R     1
203.0.113.226    12386   203.0.113.226    12386    ::      0       ::       0        -
```

Beispiel 2: vEdge1 sieht den TLOC nicht aus Farbblau, der von vEdge2 kommt. Er sieht nur MPLS-TLOC.

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


PUBLIC           PRIVATE
ADDRESS                                                                    PSEUDO
PUBLIC                    PRIVATE  PUBLIC  IPV6    PRIVATE  IPV6     BFD
FAMILY   TLOC IP          COLOR            ENCAP  FROM PEER        STATUS    KEY     PUBLIC IP
PORT    PRIVATE IP       PORT     IPV6    PORT    IPV6     PORT     STATUS
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
ipv4     1.1.1.10         mpls             ipsec  0.0.0.0          C,Red,R   1       10.20.67.10
12406   10.20.67.10      12406    ::      0       ::       0        up
         1.1.1.10         gold             ipsec  0.0.0.0          C,Red,R   1
203.0.113.225    4501    10.19.145.2      12386    ::      0       ::       0        up
         1.1.1.20         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.20
12386   10.20.67.20      12386    ::      0       ::       0        up
         1.1.1.30         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.30
12346   10.20.67.30      12346    ::      0       ::       0        up
         1.1.1.30         gold             ipsec  1.1.1.3          C,I,R     1       192.0.2.129
12386   192.0.2.129      12386    ::      0       ::       0        up
         1.1.1.40         mpls             ipsec  1.1.1.3          C,I,R     1       10.20.67.40
12426   10.20.67.40      12426    ::      0       ::       0        up
         1.1.1.40         gold             ipsec  1.1.1.3          C,I,R     1
203.0.113.226    12386   203.0.113.226    12386    ::      0       ::       0        up
```

Wenn Sie die Richtlinie überprüfen, sehen Sie, warum der TLOC nicht im vEdge1 angezeigt wird.

```
vSmart1# show running-config policy
policy
 lists
  tloc-list SITE20
   tloc 1.1.1.20 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
 !
!
```

Erkennung von bidirektionaler Weiterleitung

Den Befehl show bfd sessions verstehen

In der Ausgabe sind folgende wichtige Punkte zu beachten:

```
vEdge-2# show bfd sessions
                                      SOURCE TLOC      REMOTE TLOC
DST PUBLIC                       DST PUBLIC         DETECT      TX
SYSTEM IP        SITE ID  STATE       COLOR            COLOR           SOURCE IP
IP                               PORT        ENCAP  MULTIPLIER  INTERVAL(msec) UPTIME
TRANSITIONS
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-------------
1.1.1.10         10       down        blue             gold             10.19.146.2
203.0.113.225                    4501        ipsec  7           1000           NA              7
1.1.1.30         30       up          blue             gold             10.19.146.2
192.0.2.129                      12386       ipsec  7           1000           0:00:00:22      2
1.1.1.40         40       up          blue             gold             10.19.146.2
203.0.113.226                    12386       ipsec  7           1000           0:00:00:22      1
1.1.1.40         40       up          mpls             mpls
10.20.67.10                      10.20.67.40                      12426       ipsec  7
1000           0:00:10:11      0
```

- SYSTEM-IP: Peers, System-IP
- QUELLE- UND REMOTE-TLOC-FARBE: Dies ist hilfreich, um zu wissen, welche TLOC Sie empfangen und senden möchten.
- QUELL-IP: Es ist die private Quell-IP. Wenn Sie sich hinter einer NAT befinden, werden diese Informationen hier nicht angezeigt (dies ist mit der Verwendung von show control local-properties <wan-interface-list> erkennbar, die zu Beginn des Dokuments erläutert wird).
- DST PUBLIC IP: Es ist das Ziel, das der vEdge verwendet, um den Datenebenentunnel zu bilden, unabhängig davon, ob dieser hinter NAT liegt oder nicht. (Beispiel: Direkt mit dem Internet verbundene vEdges oder MPLS-Verbindungen (Multi-Protocol Label Switching)
- DST PUBLIC PORT: Öffentlicher NAT-basierter Port, den der vEdge verwendet, um den Datenebenentunnel zum Remote-vEdge zu bilden.
- ÜBERGÄNGE: Anzahl der Änderungen des Status der BFD-Sitzung von "NA" in "UP" und umgekehrt.

Befehls-Tunnelstatistik

Die Tunnelstatistik kann Informationen über die Datenebenentunnel anzeigen. Sie können problemlos erkennen, ob Sie Pakete für einen bestimmten IPSEC-Tunnel zwischen den vEdges senden oder empfangen. Dies kann Ihnen dabei helfen, zu verstehen, ob Pakete an jedem Ende vorhanden sind, und Verbindungsprobleme zwischen den Knoten zu isolieren.

Wenn Sie im Beispiel den Befehl mehrmals ausführen, können Sie eine Erhöhung oder keine Erhöhung der tx-pkts oder rx-pkts bemerken.

Tipp: Wenn der Zähler für tx-pkts increment (Schrittweise Erhöhung) verwendet wird, übertragen Sie Daten an den Peer. Wenn Ihre rx-pkts nicht inkrementiert werden, bedeutet dies, dass Sie keine Daten von Ihrem Peer erhalten. Überprüfen Sie in diesem Fall das andere Ende, und überprüfen Sie, ob die tx-pkts erhöht werden.

```
TCP
vEdge2# show tunnel statistics

TUNNEL SOURCE DEST TUNNEL MSS PROTOCOL SOURCE IP DEST IP PORT PORT SYSTEM IP LOCAL COLOR REMOTE
COLOR MTU tx-pkts tx-octets rx-pkts rx-octets ADJUST ----------------------------------------
-----------------------------------------------------------------------------------------------
-------------------- ipsec     172.16.16.147  10.88.244.181   12386   12406  1.1.1.10
public-internet  default          1441    38282    5904968    38276    6440071    1361
ipsec     172.16.16.147  10.152.201.104  12386   63364  100.1.1.100  public-internet  default
1441    33421    5158814    33416    5623178    1361
ipsec     172.16.16.147  10.152.204.31   12386   58851  1.1.1.90     public-internet  public-
internet  1441    12746    1975022    12744    2151926    1361
ipsec     172.24.90.129  10.88.244.181   12426   12406  1.1.1.10     biz-internet     default
1441    38293    5906238    38288    6454580    1361
ipsec     172.24.90.129  10.152.201.104  12426   63364  100.1.1.100  biz-internet     default
1441    33415    5157914    33404    5621168    1361
ipsec     172.24.90.129  10.152.204.31   12426   58851  1.1.1.90     biz-internet     public-
internet  1441    12750    1975622    12747    2152446    1361

TUNNEL                                    SOURCE
DEST
TUNNEL                                          MSS
PROTOCOL  SOURCE IP      DEST IP         PORT    PORT   SYSTEM IP    LOCAL COLOR      REMOTE
COLOR     MTU     tx-pkts  tx-octets  rx-pkts  rx-octets  ADJUST
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------
ipsec     172.16.16.147  10.88.244.181   12386   12406  1.1.1.10     public-internet
default          1441    39028    6020779    39022    6566326    1361
ipsec     172.16.16.147  10.152.201.104  12386   63364  100.1.1.100  public-internet
default          1441    34167    5274625    34162    5749433    1361
ipsec     172.16.16.147  10.152.204.31   12386   58851  1.1.1.90     public-internet  public-
internet  1441    13489    2089069    13487    2276382    1361
ipsec     172.24.90.129  10.88.244.181   12426   12406  1.1.1.10     biz-internet
default          1441    39039    6022049    39034    6580835    1361
ipsec     172.24.90.129  10.152.201.104  12426   63364  100.1.1.100  biz-internet
default          1441    34161    5273725    34149    5747259    1361
ipsec     172.24.90.129  10.152.204.31   12426   58851  1.1.1.90     biz-internet     public-
internet  1441    13493    2089669    13490    2276902    1361
```

Ein weiterer nützlicher Befehl ist **show tunnel statistics bfd**, der verwendet werden kann, um die Anzahl der BFD-Pakete zu überprüfen, die innerhalb eines bestimmten Datenebenentunnels gesendet und empfangen wurden:

```
vEdge1# show tunnel statistics bfd

BFD   BFD   BFD     BFD
                                                      BFD      BFD
PMTU  PMTU  PMTU    PMTU
TUNNEL                                  SOURCE  DEST   ECHO TX  ECHO RX  BFD ECHO   BFD ECHO
TX    RX    TX      RX
PROTOCOL  SOURCE IP      DEST IP        PORT    PORT   PKTS     PKTS     TX OCTETS  RX OCTETS
```

```
PKTS  PKTS  OCTETS  OCTETS
-------------------------------------------------------------------------------------------------
--------------------------
ipsec     192.168.109.4  192.168.109.5  4500    4500   0        0        0          0          0
0     0       0
ipsec     192.168.109.4  192.168.109.5  12346   12366  1112255  1112253  186302716  186302381
487   487   395939  397783
ipsec     192.168.109.4  192.168.109.7  12346   12346  1112254  1112252  186302552  186302210
487   487   395939  397783
ipsec     192.168.109.4  192.168.110.5  12346   12366  1112255  1112253  186302716  186302381
487   487   395939  397783
```

Zugriffsliste

Eine Zugriffsliste ist ein nützlicher und notwendiger Schritt nach dem Betrachten der Ausgabe von **show bfd sessions**. Nachdem die privaten und öffentlichen IPs und Ports bekannt sind, können Sie eine Zugriffssteuerungsliste (ACL) erstellen, die mit SRC_PORT, DST_PORT, SRC_IP und DST_IP übereinstimmt. So können Sie feststellen, ob Sie BFD-Nachrichten empfangen und senden oder nicht.

Hier finden Sie ein Beispiel für eine ACL-Konfiguration:

```
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
!
!
!
```

Im Beispiel verwendet diese ACL zwei Sequenzen. Die Sequenz 10 entspricht den BFD-Nachrichten, die von diesem vEdge an den Peer gesendet werden. Sequence 20 tut das Gegenteil.

Er wird mit dem Quell- (**Private**) und den Ziel-Ports (**Public**) verglichen. Wenn der vEdge NAT verwendet, stellen Sie sicher, dass Sie die richtigen Quell- und Zielports überprüfen.

Um die Treffer auf jedem Sequenzzähler zu überprüfen, geben Sie die **Indikatoren für die Anzeige von Richtlinien-Zugriffslisten <Name der Zugriffsliste> ein.**

```
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

Verwendung von Tools im Stun-Client zum Erkennen von NAT-Zuordnung und -Filterung

Wenn Sie alle genannten Schritte ausgeführt haben und sich hinter NAT befinden, müssen Sie im nächsten Schritt das Zuordnungs- und Filterverhalten für UDP NAT Traversal (RFC 4787) ermitteln. Dieses Tool ist wirklich hilfreich, um die lokale externe vEdge-IP-Adresse zu ermitteln, wenn sich dieser vEdge hinter einem NAT-Gerät befindet. Dieser Befehl ruft eine Port-Zuordnung für das Gerät ab und erkennt optional Eigenschaften für die NAT zwischen dem lokalen Gerät und einem Server (öffentlicher Server: Beispiel Google-Betäubungsserver).

> **Hinweis**: Weitere Informationen finden Sie unter: Docs Viptela - STUN-Client

```
vEdge1# tools stun-client vpn 0 options "--mode full --localaddr 192.168.12.100 12386 --
verbosity 2 stun.l.google.com 19302"
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
```

Bei neueren Softwareversionen kann die Syntax sehr unterschiedlich sein:

```
vEdge1# tools stun-client vpn 0 options "--mode full --localaddr 192.168.12.100 --localport
12386 --verbosity 2 stun.l.google.com 19302"
```

In diesem Beispiel führen Sie einen vollständigen NAT-Erkennungstest durch, indem Sie den UDP-Quellport 12386 zum Google STUN-Server verwenden. Die Ausgabe dieses Befehls gibt Ihnen das NAT-Verhalten und den NAT-Filtertyp basierend auf RFC 4787.

> **Hinweis**: Wenn Sie **Tools betreiben**, denken Sie daran, den STUN-Dienst in der Tunnelschnittstelle zuzulassen, da er andernfalls nicht funktioniert. Verwenden Sie **allow-service-Betäubung**, um die Betäubungsdaten übergeben zu lassen.

```
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
```

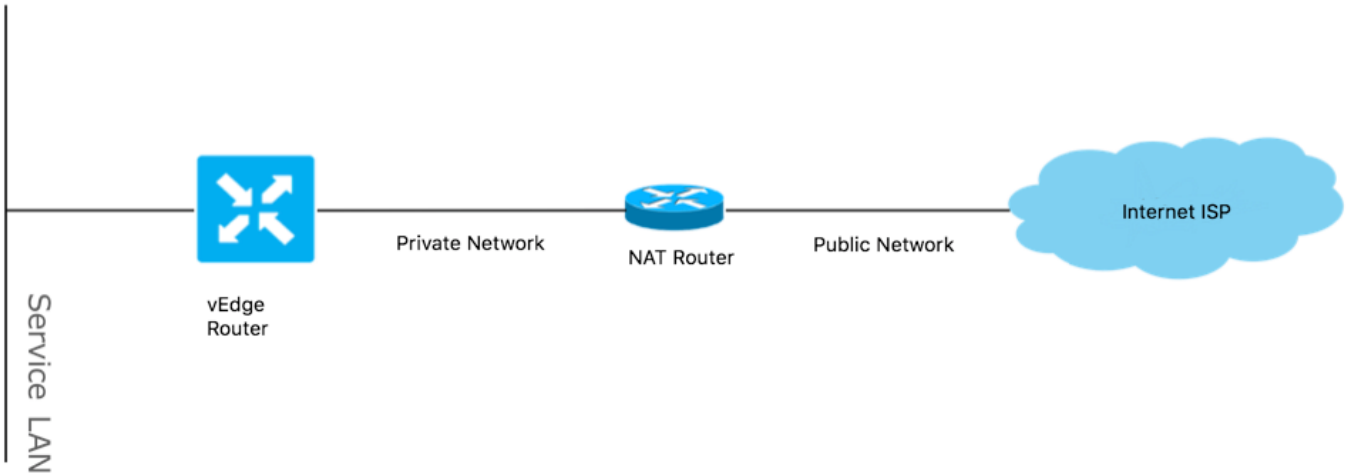
```
   color gold
   max-control-connections 1
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   no allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   allow-service stun
  !
  no shutdown
 !
!
```

Dies zeigt die Zuordnung zwischen STUN-Terminologie (Full-Cone NAT) und RFC 4787 (NAT Behavioral für 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

Unterstützte NAT-Typen für Datenebenen-Tunnel

In den meisten Fällen können Ihre öffentlichen Farben wie Biz-Internet oder öffentliches Internet direkt mit dem Internet verbunden werden. In anderen Fällen befindet sich hinter der vEdge WAN-Schnittstelle ein NAT-Gerät, und der eigentliche Internetdienstanbieter kann über eine private IP verfügen, und das andere Gerät (Router, Firewall usw.) kann das Gerät mit den öffentlich zugänglichen IP-Adressen sein.



Wenn Sie einen falschen NAT-Typ haben, kann dies einer der häufigsten Gründe sein, die die Bildung von Datenebenentunneln nicht zulassen. Dies sind die unterstützten NAT-Typen.

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**

Firewalls

Wenn Sie die NAT bereits überprüft haben und diese nicht in den nicht unterstützten Source- und Destination-Typen enthalten ist, kann es sein, dass eine Firewall die Ports blockiert, die zur Bildung der Datenebenentunnels verwendet werden.

Stellen Sie sicher, dass diese Ports in der Firewall für Datenebenenverbindungen geöffnet sind: Datenebene vEdge-zu-vEdge:

UDP 12346 bis 13156

Für Steuerungsverbindungen vom vEdge zu Controllern:

UDP 12346 bis 13156

TCP

Stellen Sie sicher, dass Sie diese Ports öffnen, um eine erfolgreiche Verbindung der Datenebenentunnel zu erreichen.

Wenn Sie die Quell- und Ziel-Ports überprüfen, die für Datenebenen-Tunnel verwendet werden, können Sie **Tunnelstatistiken anzeigen** oder **BFD-Sitzungen anzeigen |-Registerkarte**, aber keine **bfd-Sitzungen anzeigen.** Es werden keine Quellports, sondern nur Zielports angezeigt, wie Sie sehen können:

```
vEdge1# show bfd sessions
                                       SOURCE TLOC      REMOTE TLOC
DST PUBLIC                        DST PUBLIC         DETECT      TX
SYSTEM IP        SITE ID  STATE       COLOR            COLOR           SOURCE IP
IP                                 PORT        ENCAP  MULTIPLIER  INTERVAL(msec) UPTIME
TRANSITIONS
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
```

```
-------------
192.168.30.105   50       up          biz-internet     biz-internet     192.168.109.181
192.168.109.182                12346       ipsec  7           1000           1:21:28:05      10
192.168.30.105   50       up          private1         private1         192.168.110.181
192.168.110.182                12346       ipsec  7           1000           1:21:26:13      2

vEdge1# show bfd sessions | tab

                                       SRC    DST                    SITE
DETECT      TX
SRC IP           DST IP           PROTO  PORT   PORT   SYSTEM IP       ID    LOCAL COLOR   COLOR
STATE  MULTIPLIER  INTERVAL  UPTIME      TRANSITIONS
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------
192.168.109.181  192.168.109.182  ipsec  12346  12346  192.168.30.105  50    biz-internet  biz-
internet  up     7           1000      1:21:28:05  10
192.168.110.181  192.168.110.182  ipsec  12346  12346  192.168.30.105  50    private1
private1      up     7           1000      1:21:26:13  2
```

Hinweis: Weitere Informationen zu den verwendeten SD-WAN-Firewall-Ports finden Sie hier.

Sicherheit

Wenn Sie feststellen, dass Ihr ACL-Zähler sowohl ein- als auch ausgehend ansteigt, überprüfen Sie, ob mehrere Iterationen **Systemstatistiken voneinander abweichen** und sicherstellen, dass keine Verwerfungen auftreten.

```
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 dieser Ausgabe erhöht **rx_replay_integer_drop** mit jeder Iteration des Befehls **show system statistics diff**.

```
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
```

Führen Sie zunächst einen **Request Security ipsec-**rekey auf dem vEdge durch. Gehen Sie dann durch mehrere Iterationen von **show system statistics diff** und sehen Sie, ob Sie noch **rx_replay_integer_drop** sehen. Wenn ja, überprüfen Sie Ihre Sicherheitskonfiguration.

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

Wenn Sie die genannte Konfiguration haben, versuchen Sie, **ah-no-id** unter ipsec zum Authentifizierungstyp hinzuzufügen.

```
vEdge1# show running-config security
security
ipsec
authentication-type sha1-hmac ah-sha1-hmac ah-no-id
!
!
```

> **Tipp**: ah-no-id aktiviert eine geänderte Version von AH-SHA1 HMAC und ESP HMAC-SHA1, die das ID-Feld im äußeren IP-Header des Pakets ignoriert. Diese Option unterstützt einige Nicht-VIP-Geräte, wie z. B. die Apple AirPort Express NAT, die einen Fehler aufweist, der dazu führt, dass das ID-Feld im IP-Header, ein nicht mutbares Feld, geändert wird. Konfigurieren Sie die Option ah-no-id in der Liste der Authentifizierungstypen so, dass die Viptela AH-Software das ID-Feld im IP-Header ignoriert, sodass die Viptela-Software mit diesen Geräten zusammenarbeiten kann.

ISP-Probleme mit DSCP-markiertem Datenverkehr

Standardmäßig wird der gesamte Kontroll- und Verwaltungsdatenverkehr vom vEdge-Router zu den Controllern über DTLS- oder TLS-Verbindungen übertragen und mit dem DSCP-Wert CS6 (48 Dezimalstellen) gekennzeichnet. Für den Tunnelverkehr am Datenplatz verwenden vEdge-Router die IPsec- oder GRE-Kapselung, um Datenverkehr untereinander zu senden. Zur Fehlererkennung und Leistungsmessung auf Datenebene senden Router regelmäßig andere BFD-Pakete. Diese BFD-Pakete sind außerdem mit dem DSCP-Wert CS6 (48 Dezimalstellen) gekennzeichnet.

Aus ISP-Sicht werden diese Datenverkehrsarten auch als UDP-Datenverkehr mit dem DSCP-Wert CS6 angesehen, da vEdge-Router und SD-WAN-Controller standardmäßig DSCP kopieren, das als Markierung in den äußeren IP-Header dient.

So könnte es aussehen, wenn tcpdump auf einem ISP-Router für den Transit ausgeführt wird:

```
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
```

Wie hier zu sehen ist, sind alle Pakete mit dem TOS-Byte 0xc0 gekennzeichnet, auch bekannt als DS-Feld (entspricht dem Dezimalwert 192 oder 110 00 00 im Binärformat). Die ersten 6 High-

Order-Bits entsprechen dem DSCP-Bit-Wert 48 im Dezimalformat oder CS6).

Die ersten zwei Pakete in der Ausgabe entsprechen einem Kontrollebenentunnel und den beiden verbleibenden, einem Datenebenen-Tunnelverkehr. Basierend auf der Paketlänge und der TOS-Markierung kann daraus mit hoher Sicherheit schließen, dass es sich um BFD-Pakete handelt (RX- und TX-Richtung). Diese Pakete sind auch mit CS6 gekennzeichnet.

Manche Service Provider, insbesondere Anbieter von MPLS L3-VPN-/MPLS-L2-VPN-Services, unterhalten in manchen Fällenunterschiedliche SLA mit dem Kunden und kann eine andere Datenverkehrsklasse basierend auf der DSCP-Markierung des Kunden unterschiedlich behandeln. So können Sie beispielsweise über Premium-Services den DSCP EF- und CS6-Sprach- und Signalisierungsverkehr priorisieren. Da Prioritätsdatenverkehr fast immer überwacht wird, selbst wenn die Gesamtbandbreite eines Uplink nicht überschritten wird, kann dieser Datenpaketverlust erkannt werden, sodass auch BFD-Sitzungen flapping sein können.

In einigen Fällen wurde festgestellt, dass bei einer Beeinträchtigung der dedizierten Prioritätswarteschlange des Dienstanbieter-Routers keine Unterbrechungen für normalen Datenverkehr (z. B. Ausführung eines einfachen **Pings** vom vEdge-Router) auftreten, da dieser Datenverkehr mit dem DSCP-Standardwert 0 markiert ist (TOS-Byte):

```
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
```

Gleichzeitig werden Ihre BFD-Sitzungen jedoch wie folgt ablaufen:

```
show bfd history
                                                    DST PUBLIC       DST PUBLIC
RX      TX
SYSTEM IP        SITE ID  COLOR            STATE       IP               PORT        ENCAP  TIME
PKTS    PKTS    DEL
--------------------------------------------------------------------------------------------------
--------------------------------------
192.168.30.4     13       public-internet  up          192.168.109.4    12346       ipsec  2019-
05-01T03:54:23+0200 127     135     0
192.168.30.4     13       public-internet  up          192.168.109.4    12346       ipsec  2019-
05-01T03:54:23+0200 127     135     0
192.168.30.4     13       public-internet  down        192.168.109.4    12346       ipsec  2019-
05-01T03:55:28+0200 140     159     0
192.168.30.4     13       public-internet  down        192.168.109.4    12346       ipsec  2019-
05-01T03:55:28+0200 140     159     0
192.168.30.4     13       public-internet  up          192.168.109.4    12346       ipsec  2019-
05-01T03:55:40+0200 361     388     0
192.168.30.4     13       public-internet  up          192.168.109.4    12346       ipsec  2019-
05-01T03:55:40+0200 361     388     0
192.168.30.4     13       public-internet  down        192.168.109.4    12346       ipsec  2019-
05-01T03:57:38+0200 368     421     0
192.168.30.4     13       public-internet  down        192.168.109.4    12346       ipsec  2019-
05-01T03:57:38+0200 368     421     0
192.168.30.4     13       public-internet  up          192.168.109.4    12346       ipsec  2019-
05-01T03:58:05+0200 415     470     0
192.168.30.6     13       public-internet  up          192.168.109.4    12346       ipsec  2019-
```

```
05-01T03:58:05+0200 415     470     0
192.168.30.6     13       public-internet  down        192.168.109.4    12346       ipsec  2019-
05-01T03:58:25+0200 464063  464412  0
```

Und hier ist **nping** praktisch, um Probleme zu beheben:

```
vedge2# tools nping vpn 0 options "--tos 0x0c --icmp --icmp-type echo --delay 200ms -c 100 -q"
192.168.109.7
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
```

Debuggen von BFD

Wenn eine eingehendere Untersuchung erforderlich ist, sollten Sie gelegentlich das Debugging von BFD auf dem vEdge-Router ausführen. Der Forwarding Traffic Manager (FTM) ist für BFD-Vorgänge auf vEdge-Routern zuständig und muss deshalb **debug ftm bfd** werden. Alle Debugausgaben werden in **/var/log/tmplog/vdebug-**Datei gespeichert. Wenn Sie diese Meldungen auf der Konsole haben möchten (ähnlich dem Verhalten des Cisco IOS®**Terminalüberwachungssystems**), können Sie **Überwachungsstart /var/log/tmplog/vdebug** verwenden. Um die Protokollierung zu beenden, können Sie **Überwachungsstopp /var/log/tmplog/vdebug** verwenden. So sieht die Ausgabe für BFD-Sitzungen aus, die aufgrund des Timeouts ausfallen (Remote-TLOC mit IP-Adresse 192.168.110.6 ist nicht mehr erreichbar):

```
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1008]: BFD-
session TNL 192.168.110.5:12366->192.168.110.6:12346,l-tloc(32771)->r-tloc(32772),TLOC
192.168.30.5:biz-internet->192.168.30.6:public-internet IPSEC: BFD Session STATE update,
New_State :- DOWN, Reason :- LOCAL_TIMEOUT_DETECT Observed latency :- 7924, bfd_record_index :-
8, Hello timer :- 1000, Detect Multiplier :- 7
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_proc_tunnel_public_tloc_msg[252]:
tun_rec_index 13 tloc_index 32772 public tloc 0.0.0.0/0
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_increment_wanif_bfd_flap[2427]: BFD-
session TNL 192.168.110.5:12366->192.168.110.6:12346, : Increment the WAN interface counters by
1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1119]: BFD-
session TNL 192.168.110.5:12366->192.168.110.6:12346,l-tloc(32771)->r-tloc(32772),TLOC
192.168.30.5:biz-internet->192.168.30.6:public-internet IPSEC BFD session history update, old
state 3 new state 1 current  flap count 1 prev_index 1 current 2
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1140]: Attempting to add TLOC :
from_ttm 0 origin remote tloc-index 32772 pub 192.168.110.6:12346 pub v6 :::0 system_ip
192.168.30.6 color 5 spi 333
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]:
(32771:32772) proto 50 src 192.168.110.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]:
(32770:32772) proto 50 src 192.168.109.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD
session creation. Remote-tloc: tloc-index 32772, system-ip 192.168.30.6, color 5 encap 2from
local WAN Interface ge0_0
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]:
(32771:32772) proto 50 src 192.168.110.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD
```

session creation. Remote-tloc: tloc-index 32772, system-ip 192.168.30.6, color 5 encap 2from local WAN Interface ge0_1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32770:32772) proto 50 src 192.168.109.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.168.110.5:12366->192.168.110.6:12346,l-tloc(32771)->r-tloc(32772),TLOC 192.168.30.5:biz-internet->192.168.30.6:public-internet IPSEC: session sa index changed from 484 to 484
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32771:32772) src 192.168.110.5:12366 dst 192.168.110.6:12346 record index 8 ref-count 1 sa-idx 484
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.168.109.5:12366->192.168.110.6:12346,l-tloc(32770)->r-tloc(32772),TLOC 192.168.30.5:public-internet->192.168.30.6:public-internet IPSEC: session sa index changed from 485 to 485
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32770:32772) src 192.168.109.5:12366 dst 192.168.110.6:12346 record index 9 ref-count 1 sa-idx 485
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1008]: BFD-session TNL 192.168.109.5:12366->192.168.110.6:12346,l-tloc(32770)->r-tloc(32772),TLOC 192.168.30.5:public-internet->192.168.30.6:public-internet IPSEC: BFD Session STATE update, New_State :- DOWN, Reason :- LOCAL_TIMEOUT_DETECT Observed latency :- 7924, bfd_record_index :- 9, Hello timer :- 1000, Detect Multiplier :- 7
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_proc_tunnel_public_tloc_msg[252]: tun_rec_index 14 tloc_index 32772 public tloc 0.0.0.0/0
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_increment_wanif_bfd_flap[2427]: BFD-session TNL 192.168.109.5:12366->192.168.110.6:12346, : Increment the WAN interface counters by 1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1119]: BFD-session TNL 192.168.109.5:12366->192.168.110.6:12346,l-tloc(32770)->r-tloc(32772),TLOC 192.168.30.5:public-internet->192.168.30.6:public-internet IPSEC BFD session history update, old state 3 new state 1 current flap count 1 prev_index 1 current 2
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1140]: Attempting to add TLOC : from_ttm 0 origin remote tloc-index 32772 pub 192.168.110.6:12346 pub v6 :::0 system_ip 192.168.30.6 color 5 spi 333
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32771:32772) proto 50 src 192.168.110.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32770:32772) proto 50 src 192.168.109.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session creation. Remote-tloc: tloc-index 32772, system-ip 192.168.30.6, color 5 encap 2from local WAN Interface ge0_0
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32771:32772) proto 50 src 192.168.110.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD session creation. Remote-tloc: tloc-index 32772, system-ip 192.168.30.6, color 5 encap 2from local WAN Interface ge0_1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]: (32770:32772) proto 50 src 192.168.109.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.168.110.5:12366->192.168.110.6:12346,l-tloc(32771)->r-tloc(32772),TLOC 192.168.30.5:biz-internet->192.168.30.6:public-internet IPSEC: session sa index changed from 484 to 484
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32771:32772) src 192.168.110.5:12366 dst 192.168.110.6:12346 record index 8 ref-count 1 sa-idx 484
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session TNL 192.168.109.5:12366->192.168.110.6:12346,l-tloc(32770)->r-tloc(32772),TLOC 192.168.30.5:public-internet->192.168.30.6:public-internet IPSEC: session sa index changed from 485 to 485
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32770:32772) src 192.168.109.5:12366 dst 192.168.110.6:12346 record index 9 ref-count 1 sa-idx 485
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_send_bfd_msg[499]: Sending BFD notification Down notification to TLOC id 32772
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1140]: Attempting to add TLOC : from_ttm 1 origin remote tloc-index 32772 pub 192.168.110.6:12346 pub v6 :::0 system_ip 192.168.30.6 color 5 spi 333
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]: (32771:32772) proto 50 src 192.168.110.5:12366 dst 192.168.110.6:12346 ref_count 1

```
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_set_del_marker_internal[852]:
(32770:32772) proto 50 src 192.168.109.5:12366 dst 192.168.110.6:12346 ref_count 1
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 creation. Remote-tloc: tloc-index 32772, system-ip 192.168.30.6, color 5 encap 2from
local WAN Interface ge0_0
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]:
(32771:32772) proto 50 src 192.168.110.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_create[238]: Attempting BFD
session creation. Remote-tloc: tloc-index 32772, system-ip 192.168.30.6, color 5 encap 2from
local WAN Interface ge0_1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_clear_delete_marker[828]:
(32770:32772) proto 50 src 192.168.109.5:12366 dst 192.168.110.6:12346 ref_count 1
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session
TNL 192.168.110.5:12366->192.168.110.6:12346,l-tloc(32771)->r-tloc(32772),TLOC 192.168.30.5:biz-
internet->192.168.30.6:public-internet IPSEC: session sa index changed from 484 to 484
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32771:32772) src
192.168.110.5:12366 dst 192.168.110.6:12346 record index 8 ref-count 1 sa-idx 484
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_sa[1207]: BFD-session
TNL 192.168.109.5:12366->192.168.110.6:12346,l-tloc(32770)->r-tloc(32772),TLOC
192.168.30.5:public-internet->192.168.30.6:public-internet IPSEC: session sa index changed from
485 to 485
log:local7.debug: May  7 16:23:09 vedge2 FTMD[674]: ftm_tloc_add[1653]: BFD (32770:32772) src
192.168.109.5:12366 dst 192.168.110.6:12346 record index 9 ref-count 1 sa-idx 485
log:local7.info: May  7 16:23:09 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002:
Notification: 5/7/2019 14:23:9 bfd-state-change severity-level:major host-name:"vedge2" system-
ip:192.168.30.5 src-ip:192.168.110.5 dst-ip:192.168.110.6 proto:ipsec src-port:12366 dst-
port:12346 local-system-ip:192.168.30.5 local-color:"biz-internet" remote-system-ip:192.168.30.6
remote-color:"public-internet" new-state:down deleted:false flap-reason:timeout
log:local7.info: May  7 16:23:09 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002:
Notification: 5/7/2019 14:23:9 bfd-state-change severity-level:major host-name:"vedge2" system-
ip:192.168.30.5 src-ip:192.168.109.5 dst-ip:192.168.110.6 proto:ipsec src-port:12366 dst-
port:12346 local-system-ip:192.168.30.5 local-color:"public-internet" remote-system-
ip:192.168.30.6 remote-color:"public-internet" new-state:down deleted:false flap-reason:timeout
```

Ein weiterer hilfreicher Debugger zur Aktivierung ist das Debuggen von TTM-Ereignissen (Tunnel Traffic Manager), **das Debuggen von ttm-Ereignissen**. So sieht das BFD-DOWN-Ereignis aus Sicht von TTM aus:

```
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM
Msg LINK_BFD, Client: ftmd, AF: LINK
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[413]:     Remote-
TLOC: 192.168.30.6 : public-internet : ipsec, Local-TLOC: 192.168.30.5 : biz-internet : ipsec,
Status: DOWN, Rec Idx: 13 MTU: 1441, Loss: 77, Latency: 0, Jitter: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM
Msg LINK_BFD, Client: ftmd, AF: LINK
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[413]:     Remote-
TLOC: 192.168.30.6 : public-internet : ipsec, Local-TLOC: 192.168.30.5 : public-internet :
ipsec, Status: DOWN, Rec Idx: 14 MTU: 1441, Loss: 77, Latency: 0, Jitter: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM
Msg BFD, Client: ftmd, AF: TLOC-IPV4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[402]:     TLOC:
192.168.30.6 : public-internet : ipsec, Status: DOWN
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_af_tloc_db_bfd_status[234]:   BFD
message: I SAY WHAT WHAT tloc 192.168.30.6 : public-internet : ipsec status is 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Sent TTM Msg
TLOC_ADD, Client: ompd, AF: TLOC-IPV4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[213]:     TLOC:
192.168.30.6 : public-internet : ipsec, Index: 32772, Origin: REMOTE, Status: DOWN, LR enabled:
0, LR hold time: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[217]:
```

```
Attributes: GROUP PREF WEIGHT GEN-ID VERSION TLOCv4-PUB TLOCv4-PRI TLOCv6-PUB TLOCv6-PRI SITE-ID
CARRIER ENCAP RESTRICT
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:
2147483661
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: 192.168.110.6:12346
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[273]:         TLOCv4-
Private: 192.168.110.6:12346
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[277]:         TLOCv6-
Public: :::0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[281]:         TLOCv6-
Private: :::0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[285]:         TLOC-
Encap: ipsec-tunnel
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[295]:
Authentication: unknown(0x98) Encryption: aes256(0xc) SPI 334 Proto ESP
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[312]:           SPI
334, Flags 0x1e           Integrity: 1, encrypt-keys: 1 auth-keys: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[317]:
Number of protocols 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[328]:
Number of encrypt types: 2
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:
Encrypt type[0] AES256-GCM
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:
Encrypt type[1] AES256-CBC
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[339]:
Number of integrity types: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[344]:
integrity type[0] HMAC_SHA1
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, Client: ftmd, AF: TLOC-IPV4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[213]:     TLOC:
192.168.30.6 : public-internet : ipsec, Index: 32772, Origin: REMOTE, Status: DOWN, LR enabled:
0, LR hold time: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[217]:
Attributes: GROUP PREF WEIGHT GEN-ID VERSION TLOCv4-PUB TLOCv4-PRI TLOCv6-PUB TLOCv6-PRI SITE-ID
CARRIER ENCAP RESTRICT
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:
2147483661
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: 192.168.110.6:12346
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[273]:         TLOCv4-
Private: 192.168.110.6:12346
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[277]:         TLOCv6-
Public: :::0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[281]:         TLOCv6-
Private: :::0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[285]:         TLOC-
Encap: ipsec-tunnel
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[295]:
Authentication: unknown(0x98) Encryption: aes256(0xc) SPI 334 Proto ESP
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[312]:           SPI
334, Flags 0x1e           Integrity: 1, encrypt-keys: 1 auth-keys: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[317]:
Number of protocols 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[328]:
Number of encrypt types: 2
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:
Encrypt type[0] AES256-GCM
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:
Encrypt type[1] AES256-CBC
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[339]:
Number of integrity types: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[344]:
integrity type[0] HMAC_SHA1
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, Client: fpmd, AF: TLOC-IPV4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[213]:     TLOC:
192.168.30.6 : public-internet : ipsec, Index: 32772, Origin: REMOTE, Status: DOWN, LR enabled:
0, LR hold time: 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[217]:
Attributes: GROUP PREF WEIGHT GEN-ID VERSION TLOCv4-PUB TLOCv4-PRI TLOCv6-PUB TLOCv6-PRI SITE-ID
CARRIER ENCAP RESTRICT
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:
2147483661
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: 192.168.110.6:12346
```

```
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[273]:         TLOCv4-
Private: 192.168.110.6:12346
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[277]:         TLOCv6-
Public: :::0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[281]:         TLOCv6-
Private: :::0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[285]:         TLOC-
Encap: ipsec-tunnel
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[295]:
Authentication: unknown(0x98) Encryption: aes256(0xc) SPI 334 Proto ESP
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[312]:           SPI
334, Flags 0x1e           Integrity: 1, encrypt-keys: 1 auth-keys: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[317]:
Number of protocols 0
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[328]:
Number of encrypt types: 2
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:
Encrypt type[0] AES256-GCM
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[333]:
Encrypt type[1] AES256-CBC
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[339]:
Number of integrity types: 1
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[344]:
integrity type[0] HMAC_SHA1
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_DEVICE_ADD, Client: pimd, AF: DATA-DEVICE-IPV4
log:local7.debug: May  7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[431]:     Device:
192.168.30.6, Status: 2
log:local7.info: May  7 16:58:19 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002:
Notification: 5/7/2019 14:58:19 bfd-state-change severity-level:major host-name:"vedge2" system-
ip:192.168.30.5 src-ip:192.168.110.5 dst-ip:192.168.110.6 proto:ipsec src-port:12366 dst-
port:12346 local-system-ip:192.168.30.5 local-color:"biz-internet" remote-system-ip:192.168.30.6
remote-color:"public-internet" new-state:down deleted:false flap-reason:timeout
log:local7.info: May  7 16:58:20 vedge2 FTMD[674]: %Viptela-vedge2-ftmd-6-INFO-1400002:
Notification: 5/7/2019 14:58:19 bfd-state-change severity-level:major host-name:"vedge2" system-
ip:192.168.30.5 src-ip:192.168.109.5 dst-ip:192.168.110.6 proto:ipsec src-port:12366 dst-
port:12346 local-system-ip:192.168.30.5 local-color:"public-internet" remote-system-
ip:192.168.30.6 remote-color:"public-internet" new-state:down deleted:false flap-reason:timeout
```

Zugehörige Informationen

- SDWAN-Produktdokumentation
- Anatomie: Ein Blick in Network Address Translators
- Technischer Support und Dokumentation - Cisco Systems