SD-WAN - Troubleshoot GRE Interface Issues

Contents

Introduction

This document describes how to troubleshoot Generic Routing Encapsulation (GRE) interface issues in an SD-WAN environment.

Background Information

In Cisco Viptela Solution, the use cases for GRE Interfaces include:

- Send traffic to ZScaler (HTTP-Proxy) via vSmart Data-Policy or locally.
- Primary service GRE interface with default-back-up to the data center.
- Service chaining

There are cases, when the GRE interface may not come up and/or not working.

In those situations, check for

- GRE Interface is up/up via: show interface gre*
- GRE Keepalives via: show tunnel gre-keepalives

Methodology

If there is an issue, configure an Access Control List (ACL or access-list) to see if the GRE (47) packets are going out/in.

You are unable to see the GRE packets via TCP Dump, as the packets are generated by the fast path.

Sometimes, because of network address translation (NAT), GRE Keepalives can be dropped. In this case, disable the keepalive and see if the tunnel comes up.

Also, if the GRE Tunnel is constantly flapping and disabling keepalives, this keeps the interface up/up.

However, it has a drawback, where if there is a legitimate issue, it is hard to find out that GRE does not work.

See here in the document which shows an example.

This is a working GRE interface config

IN VPN0

```
vpn 0
 interface gre1
  ip address 192.0.2.1/30
  tunnel-source      <SRC-IP>
  tunnel-destination <DST-IP>
  tcp-mss-adjust     1300
  no shutdown
 !
 interface gre2
  ip address 192.0.2.5/30
  tunnel-source      <SRC-IP>
  tunnel-destination <DST-IP>
  tcp-mss-adjust     1300
  no shutdown
 !
!
```

IN Service side

```
vpn <SRVC-VPN>
service FW interface gre1 gre2
```

In Cisco SD-WAN solution based on vEdge routes, GRE interfaces working as Active-standby and not Active-Active.

At any given time, there is only GRE Interface which is in Up/Up state.

Practice

Create a policy for access-lists

```
vEdge# show running-config policy access-list
policy
 access-list GRE-In
  sequence 10
   match
    protocol 47
   !
   action accept
    count gre-in
   !
  !
  default-action accept
 !
 access-list GRE-Out
  sequence 10
   match
    protocol 47
   !
   action accept
    count gre-out
   !
  !
  default-action accept
 !
!
vEdge#
```

Create counters **gre-in** and **gre-out** and then you need to apply ACL to the interface (our tunnel rides over ge0/0).

The above ACL can be applied with the source address of the physical interface and destination address of the GRE endpoint.

```
vEdge# show running-config vpn 0 interface ge0/0
vpn 0
 interface ge0/0
  ip address 198.51.100.1/24
  tunnel-interface
   encapsulation ipsec
   max-control-connections 1
   allow-service all
   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
  access-list GRE-In in
  access-list GRE-Out out
 !
!
vEdge#
```

Now you can see the counters for GRE packets in and out because these are in the fast path, one cannot see with **tcpdump** utility.

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

         COUNTER
NAME     NAME     PACKETS  BYTES
----------------------------------
GRE-In   gre-in   176      10736
GRE-Out  gre-out  88       2112

vEdge#
```

This is our GRE tunnel.

```
vEdge# show interface gre1

                                  IF      IF      IF
TCP
                AF                ADMIN   OPER    TRACKER  ENCAP  PORT
SPEED          MSS                 RX       TX
VPN  INTERFACE  TYPE  IP ADDRESS  STATUS  STATUS  STATUS   TYPE   TYPE     MTU   HWADDR
MBPS   DUPLEX  ADJUST  UPTIME      PACKETS  PACKETS
-----------------------------------------------------------------------------------------
-----------------------------------------------------------
0    gre1      ipv4  192.0.2.1/30 Up    Up      NA       null   service  1500  05:05:05:05:00:00
```

```
1000   full    1420    0:07:10:28  2968     2968

vEdge#

vEdge# show running-config vpn 0 interface gre1
vpn 0
interface gre1
ip address 192.0.2.1/30/30
tunnel-source-interface ge0/0
tunnel-destination 192.0.2.5/30
no shutdown
!
!
vEdge#
```

You can verify if the traffic is going on the GRE interface via **show app cflowd flows** command.

This is a sample example shows bi-directional traffic (both from ingress and egress):

```
vEdge# show app cflowd flows

                                                            TCP
                                                TIME    EGRESS  INGRESS
                                  SRC   DEST          IP      CNTRL  ICMP                 TOTAL
TOTAL      MIN  MAX                             TO      INTF    INTF
VPN  SRC IP        DEST IP        PORT  PORT   DSCP  PROTO  BITS   OPCODE  NHOP IP      PKTS
BYTES      LEN  LEN   START TIME                EXPIRE  NAME    NAME
------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------
10   203.0.113.1  203.0.113.11  61478 443    0     6      16     0       203.0.113.254 3399
286304     60   1339  Sun Apr  8 10:23:05 2018  599     gre1    ge0/6
10   203.0.113.11  203.0.113.1  443   61478  0     6      24     0       203.0.113.1262556
192965     40   1340  Sun Apr  8 10:23:05 2018  592     ge0/6   gre1
```

An example of disabling keepalives (KA) on the GRE interface:

Default KA is 10 (hello-interval) and 3 (tolerance)

A KA of 0 0, disables the KA on the GRE interface.

```
vEdge# show running-config vpn 0 interface gre* | details
vpn 0
interface gre1
  description         "Primary ZEN"
  ip address <ip/mask>
keepalive 0 0
  tunnel-source       <SRC-IP-Addr>
  tunnel-destination  <DST-IP-Addr>
  no clear-dont-fragment
  mtu                 1500
  tcp-mss-adjust      1300
  no shutdown
!
```

A GRE Interface which is UP/Down shows as UP/UP (by passing the KA check).

See, TX counter here as it increases when KA is OFF. It means, vEdge is TX the packets, but you don't see the increase in RX counter, which points to a remote issue.

```
vEdge# show interface gre*
```

```
                                 IF      IF
        TCP
                                 ADMIN   OPER    ENCAP  PORT                          SPEED
        MSS                 RX         TX
VPN  INTERFACE  IP ADDRESS       STATUS  STATUS  TYPE   TYPE     MTU   HWADDR             MBPS
DUPLEX  ADJUST  UPTIME      PACKETS    PACKETS
-------------------------------------------------------------------------------------------------
---------------------------------------------------
### With KA ON
0    gre1      192.0.2.1/30  Up      Down    null   service  1500  cb:eb:98:02:00:00  -      -
    1300    -           413218129  319299248
### With KA OFF
0    gre1       192.0.2.1/30  Up      Up      null   service  1500  cb:eb:98:02:00:00  100
half    1300    0:00:01:19  413218129  319299280
```