Inappropriate Usage of "policy action set tloc-list" Leads to Traffic Blackholing

Contents

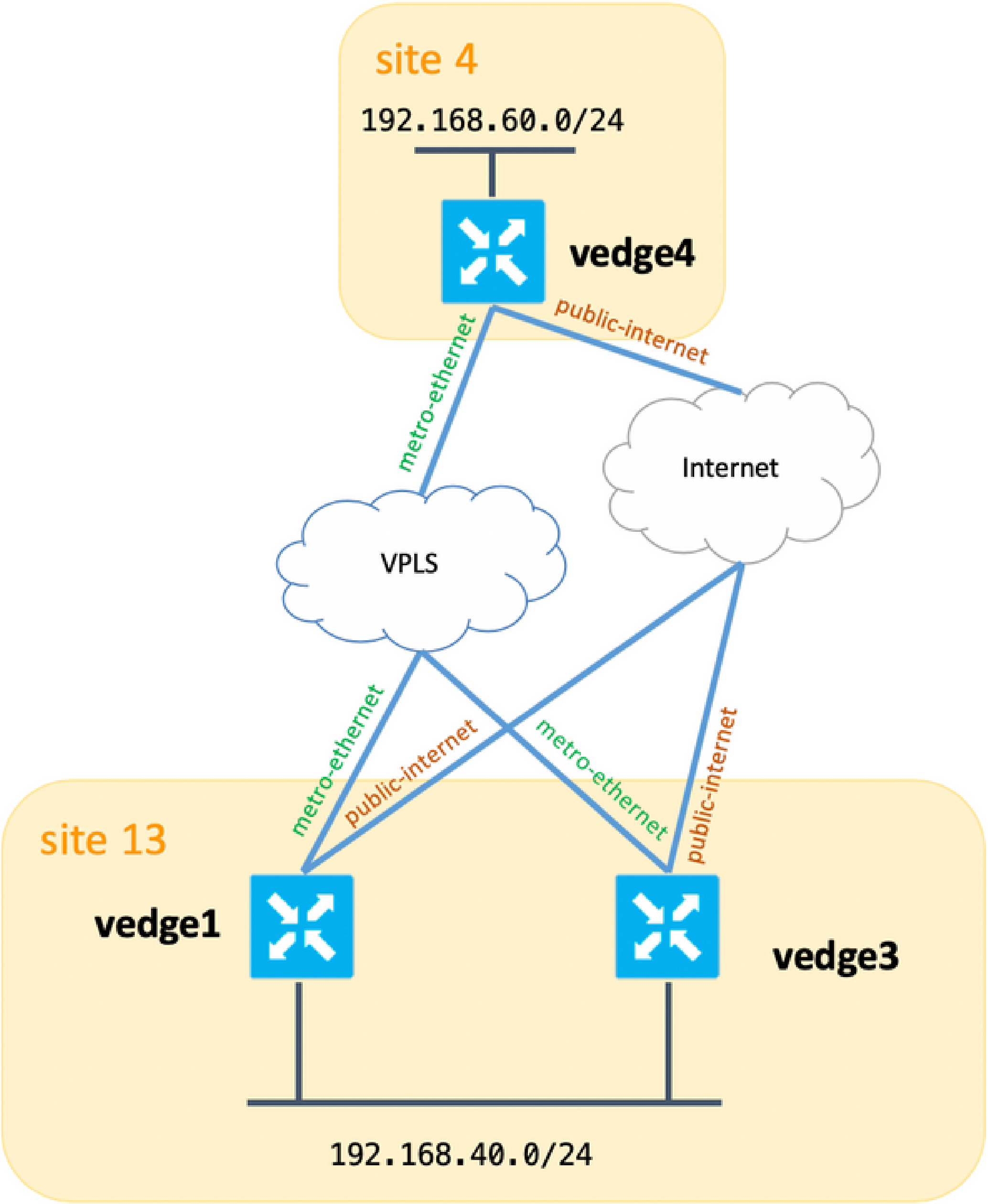
Introduction

This document describes the inappropriate policy application of **set tloc-list** action that leads to traffic blackholing in certain situations when the preferred link goes down but backup paths are still available.

> **Note**: All command outputs presented in this document are from vEdge routers. However, the troubleshooting approach remains the same for a router that runs the IOS®-XE SDWAN software. Use **sdwan** keyword in order to get same outputs on IOS®-XE SDWAN software. For example, **show sdwan omp routes** instead of **show omp routes**.

Background Information

For the purpose of demonstration and in order to better understand the problem described later, consider this topology diagram:



Besides that, here is the table that summarizes system settings:

hostname	site-id	system-ip
vedge1	13	10.155.0.118
vedge3	13	10.155.0.120
vedge4	4	10.155.0.50
vsmart1	1	10.155.0.3

Both vEdge1 and vEdge3 have a static route configured that points to some next hop in the service-side VPN:

```
vpn 40
 ip route 10.223.115.101/32 192.168.40.10
!
```

In order to achieve these goals:

1. Make vEdge1 metro-ethernet link to be preferred link for ingress traffic entering "site 13".

2. Make vEdge3 metro-ethernet link to be second preferred link for ingress traffic entering "site 13".

3. Make vEdge1 public-internet link to be third preferred link for ingress traffic entering "site 13".

4. Make vEdge3 public-internet link to be least preferred link for ingress traffic entering "site 13".

This vSmart control policy is configured:

```
policy
 lists
  tloc-list SITE13_TLOC_PREF
   tloc 10.155.0.118 color metro-ethernet encap ipsec preference 200
   tloc 10.155.0.118 color public-internet encap ipsec preference 100
   tloc 10.155.0.120 color metro-ethernet encap ipsec preference 150
   tloc 10.155.0.120 color public-internet encap ipsec preference 50
  !
  prefix-list SITE13_PREFIX
   ip-prefix 10.223.115.101/32
  !
  site-list site13
   site-id 13
  !
 control-policy TE_POLICY_2_SITE4
  sequence 10
   match route
    prefix-list SITE13_PREFIX
   !
   action accept
    set
     tloc-list SITE13_TLOC_PREF
    !
   !
  !
  default-action accept
 !
!
apply-policy
 site-list site4
  control-policy TE_POLICY_2_SITE4 out
 !
!
```

Problem

Normal Conditions

vSmart gets these routes with 4 possible TLOCs as next-hops:

```
vsmart1# show omp routes 10.223.115.101/32 | b PATH
                                            PATH                     ATTRIBUTE
VPN    PREFIX              FROM PEER        ID     LABEL    STATUS    TYPE       TLOC IP
COLOR            ENCAP  PREFERENCE
------------------------------------------------------------------------------------------------
--------------------------------------
```

```
40     10.223.115.101/32   10.155.0.118     35     1002     C,R       installed  10.155.0.118
metro-ethernet   ipsec  -
                           10.155.0.118     37     1002     C,R       installed  10.155.0.118
public-internet  ipsec  -
                           10.155.0.120     35     1002     C,R       installed  10.155.0.120
metro-ethernet   ipsec  -
                           10.155.0.120     37     1002     C,R       installed  10.155.0.120
public-internet  ipsec  -
```

And sets a preference for advertised routes accordingly:

```
vsmart1# show omp routes 10.223.115.101/32 detail | nomore | b ADVERTISED | b "peer
10.155.0.50" | i Attributes\|originator\|\ tloc\|preference
    Attributes:
     originator       10.155.0.118
     tloc             10.155.0.120, public-internet, ipsec
     preference       50
    Attributes:
     originator       10.155.0.118
     tloc             10.155.0.120, metro-ethernet, ipsec
     preference       150
    Attributes:
     originator       10.155.0.118
     tloc             10.155.0.118, public-internet, ipsec
     preference       100
    Attributes:
     originator       10.155.0.118
     tloc             10.155.0.118, metro-ethernet, ipsec
     preference       200
```

vEdge4 selects a proper TLOC and installs this route into the routing table:

```
vedge4# show ip routes 10.223.115.101/32 | b PROTOCOL
                                            PROTOCOL  NEXTHOP     NEXTHOP          NEXTHOP
VPN    PREFIX              PROTOCOL         SUB TYPE  IF NAME     ADDR             VPN      TLOC
IP          COLOR            ENCAP  STATUS
------------------------------------------------------------------------------------------------
---------------------------------------------
40     10.223.115.101/32   omp              -         -           -                -
10.155.0.118     metro-ethernet   ipsec  F,S
```

Traffic forwarding works as intended:

```
vedge4# traceroute vpn 40 10.223.115.101
Traceroute  10.223.115.101 in VPN 40
traceroute to 10.223.115.101 (10.223.115.101), 30 hops max, 60 byte packets
 1  192.168.40.4 (192.168.40.4)  0.835 ms  0.984 ms  1.097 ms
 2  192.168.40.10 (192.168.40.10)  2.955 ms  3.056 ms  3.218 ms
```

Fault Conditions

Eventually, a fault occurs on vEdge1 and the service-side LAN facing interface goes down (or is shut down by administrator in order to perform a test, for example, the result will be the same):

```
vedge1# show interface vpn 40

                                       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
------------------------------------------------------------------------------------------------
----------------------------------------------------------
40   ge0/4      ipv4  192.168.40.4/24  Up    Down    NA       null   service  1500
00:50:56:be:91:36  -      -       1420    -       129768   0
```

Because vEdge1 does not have a valid next-hop for 10.223.115.101/32 route, this route is removed from the routing and forwarding tables and does not advertise it anymore to vSmart:

```
vedge1# show ip routes 10.223.115.101/32 | b PROTO
                                            PROTOCOL  NEXTHOP     NEXTHOP          NEXTHOP
VPN    PREFIX              PROTOCOL          SUB TYPE  IF NAME     ADDR             VPN      TLOC
IP          COLOR            ENCAP  STATUS
------------------------------------------------------------------------------------------------
---------------------------------------------
40     10.223.115.101/32   static            -         -           192.168.40.21    -        -
-                -      I

vedge1# show ip fib vpn 40 | i 10.223.115.101/32
vedge1#
vedge1# show omp routes 10.223.115.101/32 detail | nomore | b ADVERTISED
vedge1#
```

At the same time, vEdge3 still advertises this route (this is expected):

```
vedge3# show omp routes 10.223.115.101/32 detail | nomore | b ADVERTISED
            ADVERTISED TO:
peer    10.155.0.3
    Attributes:
     originator       10.155.0.120
     label            1002
     path-id          35
     tloc             10.155.0.120, metro-ethernet, ipsec
     ultimate-tloc    not set
     domain-id        not set
     site-id          13
     overlay-id        1
     preference       not set
     tag              not set
     origin-proto     static
     origin-metric    0
     as-path          not set
     unknown-attr-len not set
    Attributes:
     originator       10.155.0.120
     label            1002
     path-id          37
     tloc             10.155.0.120, public-internet, ipsec
     ultimate-tloc    not set
     domain-id        not set
     site-id          13
     overlay-id        1
     preference       not set
     tag              not set
     origin-proto     static
     origin-metric    0
     as-path          not set
     unknown-attr-len not set
```

vSmart gets 2 routes now from vEdge3 as expected:

```
vsmart1# show omp routes 10.223.115.101/32 | b PATH
                                            PATH                       ATTRIBUTE
VPN    PREFIX              FROM PEER        ID     LABEL    STATUS     TYPE       TLOC IP
COLOR            ENCAP  PREFERENCE
------------------------------------------------------------------------------------------------
--------------------------------------
40     10.223.115.101/32   10.155.0.120     35     1002     C,R        installed  10.155.0.120
metro-ethernet   ipsec  -
                           10.155.0.120     37     1002     C,R        installed  10.155.0.120
public-internet  ipsec  -
```

But at the same time, vSmart continues to advertise this:

```
vsmart1# show omp routes 10.223.115.101/32 detail | nomore | b ADVERTISED | b "peer
10.155.0.50" | i Attributes\|originator\|\ tloc\|preference
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.120, public-internet, ipsec
     preference       50
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.120, metro-ethernet, ipsec
     preference       150
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.118, public-internet, ipsec
     preference       100
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.118, metro-ethernet, ipsec
     preference       200
```

As you can see, the only originator was changed and this is expected behavior because **tloc-list** action acts similar to (roughly speaking) "set next-hop" and forcefully sets the wrong TLOC, hence reachability is lost.

```
vedge4# ping vpn 40 10.223.115.101 count 5
Ping in VPN 40
PING 10.223.115.101 (10.223.115.101) 56(84) bytes of data.
^C
--- 10.223.115.101 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 3999ms

vedge4# traceroute vpn 40 10.223.115.101
Traceroute  10.223.115.101 in VPN 40
traceroute to 10.223.115.101 (10.223.115.101), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
```

Solution

As a solution, this approach is proposed in order to avoid setting the wrong TLOC next-hop information:

```
policy
 lists
  tloc-list vedge1-tlocs
   tloc 10.155.0.118 color metro-ethernet encap ipsec
   tloc 10.155.0.118 color public-internet encap ipsec
  !
  tloc-list vedge1-tlocs-preference
   tloc 10.155.0.118 color metro-ethernet encap ipsec preference 200
   tloc 10.155.0.118 color public-internet encap ipsec preference 100
  !
  tloc-list vedge3-tlocs
   tloc 10.155.0.120 color metro-ethernet encap ipsec
   tloc 10.155.0.120 color public-internet encap ipsec
  !
  tloc-list vedge3-tlocs-preference
   tloc 10.155.0.120 color metro-ethernet encap ipsec preference 150
   tloc 10.155.0.120 color public-internet encap ipsec preference 50
  !
 !
!
policy
 control-policy TE_POLICY_2_SITE4
  sequence 10
   match route
    prefix-list SITE13_PREFIX
    tloc-list   vedge1-tlocs
   !
   action accept
    set
     tloc-list vedge1-tlocs-preference
    !
   !
  !
  sequence 20
   match route
    prefix-list SITE13_PREFIX
    tloc-list   vedge3-tlocs
   !
   action accept
    set
     tloc-list vedge3-tlocs-preference
    !
   !
  !
  default-action accept
 !
!
```

Such a policy improves the situation and prevents the advertisement of the route with the wrong TLOC next-hop:

```
vsmart1# show omp routes 10.223.115.101/32 detail | nomore | b ADVERTISED | b "peer
10.155.0.50" | i Attributes\|originator\|\ tloc\|preference
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.120, public-internet, ipsec
     preference       50
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.120, metro-ethernet, ipsec
     preference       150
    Attributes:
     originator       10.155.0.120
     tloc             10.155.0.120, public-internet, ipsec
     preference       not set
```

And as a result, reachability throughout the failure scenarios is preserved:

```
vedge4# traceroute vpn 40 10.223.115.101
Traceroute  10.223.115.101 in VPN 40
traceroute to 10.223.115.101 (10.223.115.101), 30 hops max, 60 byte packets
 1  192.168.40.6 (192.168.40.6)  0.458 ms  0.507 ms  0.617 ms
 2  192.168.40.10 (192.168.40.10)  1.928 ms  1.976 ms  2.069 ms

vedge4# ping vpn 40 10.223.115.101
Ping in VPN 40
PING 10.223.115.101 (10.223.115.101) 56(84) bytes of data.
64 bytes from 10.223.115.101: icmp_seq=1 ttl=254 time=0.702 ms
64 bytes from 10.223.115.101: icmp_seq=2 ttl=254 time=0.645 ms
64 bytes from 10.223.115.101: icmp_seq=3 ttl=254 time=0.691 ms
64 bytes from 10.223.115.101: icmp_seq=4 ttl=254 time=0.715 ms
64 bytes from 10.223.115.101: icmp_seq=5 ttl=254 time=0.603 ms
^C
--- 10.223.115.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.603/0.671/0.715/0.044 ms
```