Why vEdges Unable To Establish IPSec Tunnels If NAT is being Used?

Contents

Introduction
Background information
Problem
Working Scenario
Failure Scenario
Solution
NAT Port-Forward
Explicit ACL
Other Considerations
Conclusion

Introduction

This document describes the problem that may arise when vEdge routers are using IPSec encapsulation for data plane tunnels and one device is behind Network Address Translation (NAT) device doing Symmetric NAT (RFC3489) or Address Dependent Mapping (RFC4787), while another has Direct Internet Access (DIA) or some other type of NAT configured on the transport side interface.

Background information

Note: This article is applicable for vEdge routers only and was written based on behavior seen in vEdge software 18.4.1 and 19.1.0. In newer releases behavior may be different. Please consult with documentation or contact the Cisco Technical Assistance Center (TAC) in case of doubts.

For the purpose of the demonstration, the problem was reproduced in the SD-WAN TAC lab. Devices settings are summarised in the table here:

<table>
<thead>
<tr>
<th>hostname</th>
<th>site-id</th>
<th>system-ip</th>
<th>private-ip</th>
<th>public-ip</th>
</tr>
</thead>
<tbody>
<tr>
<td>vedge1</td>
<td>232</td>
<td>10.10.10.232</td>
<td>192.168.10.232</td>
<td>198.51.100.232</td>
</tr>
<tr>
<td>vedge2</td>
<td>233</td>
<td>10.10.10.233</td>
<td>192.168.9.233</td>
<td>192.168.9.233</td>
</tr>
<tr>
<td>vsmart</td>
<td>1</td>
<td>10.10.10.228</td>
<td>192.168.0.228</td>
<td>192.168.0.228</td>
</tr>
<tr>
<td>vbond</td>
<td>1</td>
<td>10.10.10.231</td>
<td>192.168.0.231</td>
<td>192.168.0.231</td>
</tr>
</tbody>
</table>

Transport side configuration is quite generic on both devices. This is the configuration of vEdge1:
In order to demonstrate the problem in this document, Virtual Adaptive Security Appliance (ASAv) firewall resides between two vEdge routers. ASAv is doing address translations according to these rules:

- If traffic from vEdge1 is intended for controllers, source ports 12346-12426 are translated to 52346-52426
- If traffic from vEdge1 is intended for data plane connections to other sites, source ports 12346-12426 are translated to 42346-42426
- All other traffic from vEdge1 is also mapped to the same public address (198.51.100.232)

This is ASAv NAT configuration for reference:
Problem

Working Scenario

In the normal state, we can observe that data plane tunnels are established, Bidirectional Forwarding Detection (BFD) is in up state.

Please notice which public port used on vEdge1 device (52366) to establish control connections with controllers:

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ge0/0</td>
<td>198.51.100.232</td>
<td>52366</td>
<td>192.168.10.232</td>
<td>::</td>
<td>12366</td>
<td>2/1</td>
<td>biz-internet</td>
<td>up</td>
<td>2</td>
<td>no/yes/no</td>
<td>No/No</td>
<td>0:00:00:28</td>
<td>0:11:59:17</td>
<td>N</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

On vEdge2 no NAT is being used, hence private address and ports are the same:

vEdge2# 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

<table>
<thead>
<tr>
<th>INTERFACE</th>
<th>PUBLIC</th>
<th>PUBLIC PRIVATE</th>
<th>PRIVATE</th>
<th>PUBLIC</th>
<th>PUBLIC PRIVATE</th>
<th>PRIVATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
In the **show tunnel statistics** from vEdge1 we can see tx/rx counters are incrementing:

```text
vEdge1# show tunnel statistics dest-ip 192.168.9.233
```

<table>
<thead>
<tr>
<th>TCP</th>
<th>TUNNEL</th>
<th>SOURCE</th>
<th>DEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROTOCOL</td>
<td>SOURCE IP</td>
<td>DEST IP</td>
<td>PORT</td>
</tr>
<tr>
<td>ipsec</td>
<td>192.168.10.232</td>
<td>192.168.9.233</td>
<td>12366</td>
</tr>
<tr>
<td>SYSTEM IP</td>
<td>LOCAL COLOR</td>
<td>REMOTE COLOR</td>
<td></td>
</tr>
<tr>
<td>10.10.10.233</td>
<td>biz-internet</td>
<td>biz-internet</td>
<td></td>
</tr>
<tr>
<td>ADJUST</td>
<td>tx-pkts</td>
<td>tx-octets</td>
<td>rx-pkts</td>
</tr>
<tr>
<td>1441</td>
<td>223</td>
<td>81163</td>
<td>179</td>
</tr>
<tr>
<td>1202</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the same output from vEdge2 you can see as well rx/rx packets counters are incrementing. Please notice destination port (42366) is different from port used to establish control connections (52366):

```text
vEdge2# show tunnel statistics dest-ip 198.51.100.232
```

<table>
<thead>
<tr>
<th>TCP</th>
<th>TUNNEL</th>
<th>SOURCE</th>
<th>DEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROTOCOL</td>
<td>SOURCE IP</td>
<td>DEST IP</td>
<td>PORT</td>
</tr>
<tr>
<td>ipsec</td>
<td>192.168.9.233</td>
<td>198.51.100.232</td>
<td>12366</td>
</tr>
<tr>
<td>SYSTEM IP</td>
<td>LOCAL COLOR</td>
<td>REMOTE COLOR</td>
<td></td>
</tr>
<tr>
<td>10.10.10.232</td>
<td>biz-internet</td>
<td>biz-internet</td>
<td></td>
</tr>
<tr>
<td>ADJUST</td>
<td>tx-pkts</td>
<td>tx-octets</td>
<td>rx-pkts</td>
</tr>
<tr>
<td>1441</td>
<td>296</td>
<td>88669</td>
<td>261</td>
</tr>
<tr>
<td>1201</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

But BFD sessions are still up on both devices:

```text
vEdge1# show bfd sessions site-id 233 | tab
```

<table>
<thead>
<tr>
<th>DETECT</th>
<th>SRC</th>
<th>DST</th>
<th>SITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROTO</td>
<td>PORT</td>
<td>PORT</td>
<td>SYSTEM IP</td>
</tr>
<tr>
<td>STATE</td>
<td>MULTIPLIER</td>
<td>INTERVAL</td>
<td>UPTIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192.168.10.232</td>
<td>192.168.9.233</td>
<td>ipsec</td>
<td>12366</td>
</tr>
</tbody>
</table>
Different ports used for control and data plane connections does not cause any issues, connectivity is in place.

**Failure Scenario**

The user wants to enable Direct Internet Access (DIA) on vEdge2 router. In order to do so, this configuration was applied to vEdge2:

```plaintext
vpn 0
  interface ge0/1
    nat
      respond-to-ping
    !
    !
vpn 1
  ip route 0.0.0.0/0 vpn 0
```

And BFD session went down unexpectedly and moreover stays in the downstate. After clearing tunnel statistics you can see that RX counter does not increase in the `show tunnel statistics` output:

```
vEdge2# show tunnel statistics dest-ip 198.51.100.232
```

```
TCP
TUNNEL SOURCE DEST
TUNNEL PROTOCOL SOURCE IP DEST IP PORT PORT SYSTEM IP LOCAL COLOR REMOTE COLOR
MTU tx-pkts tx-octets rx-pkts rx-octets ADJUST
-----------------------------------------------------------------------------------------------
ipsec 192.168.9.233 198.51.100.232 12346 52366 10.10.10.232 232 biz-internet biz-internet 1442 282 48222 0 0 1368
```

```
vEdge2# show bfd sessions site-id 232
```

```
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
```
Initially, customer suspected that problem related to Tunnel MTU. If you compare outputs above with outputs from "Working Scenario" section, you can notice that in working scenario Tunnel MTU is 1441 versus 1442 in the failed scenario. Based on the documentation, Tunnel MTU should be 1442 (1500 default interface MTU - 58 bytes for tunnel overhead), but once BFD is up, Tunnel MTU is lowered by 1 byte. For your reference, outputs from show tunnel statistics together with show tunnel statistics bfd provided below for case when BFD is in down state:

```bash
vEdge1# show tunnel statistics dest-ip 192.168.9.233 ; show tunnel statistics bfd dest-ip 192.168.9.233
TCP
<table>
<thead>
<tr>
<th>PROTOCOL</th>
<th>SOURCE IP</th>
<th>DEST IP</th>
<th>PORT</th>
<th>PORT</th>
<th>SYSTEM IP</th>
<th>LOCAL COLOR</th>
<th>REMOTE COLOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1442</td>
<td>285</td>
<td>48735</td>
<td>0</td>
<td>0</td>
<td>1362</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BFD ECHO ECHO ECHO ECHO PMTU PMTU
TX RX TX RX TX RX

<table>
<thead>
<tr>
<th>PROTOCOL</th>
<th>SOURCE IP</th>
<th>DEST IP</th>
<th>PORT</th>
<th>PORT</th>
<th>PKTS</th>
<th>OCTETS</th>
<th>PKTS</th>
<th>OCTETS</th>
<th>PKTS</th>
<th>OCTETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipsec</td>
<td>192.168.10.232</td>
<td>192.168.9.233</td>
<td>12346</td>
<td>12346</td>
<td>133</td>
<td>22743</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
And if BFD is in up state:

vEdge1# show tunnel statistics dest-ip 192.168.9.233 ; show tunnel statistics bfd dest-ip 192.168.9.233 ;
Note: By the way, we can determine BFD packet size together with encapsulation by looking to outputs above. Note that only one BFD packet was received between two outputs, hence substracting BFD Echo RX Octets value 584987 - 584816 will give us 171-byte result. It can be useful to precisely calculate bandwidth used by BFD itself.

The reason for BFD stuck in *down* state is not MTU, but NAT configuration obviously. This is the only thing changed between *Working scenario* and *Failed scenario*. You can see here that as a result of DIA configuration, NAT static mapping was automatically created by vEdge2 in the translation table to allow data plane IPSec traffic bypass:

As you can see, port 52366 is being used instead of 42366. This is because vEdge2 expects 52366 port and learned it from OMP TLOCs advertised by vSmart:
Solution

NAT Port-Forward

From first glance, workaround for such type of problems is simple. You can configure static NAT exemption port forwarding on vEdge2 transport interface to bypass filtering for data plane connections from any sources forcefully:

```
vpn 0
interface ge0/1
nat
  respond-to-ping
  port-forward port-start 12346 port-end 12445 proto udp
  private-vpn 0
  private-ip-address 192.168.9.233
```

Here range 12346 to 12446 accommodate all possible initial ports (12346, 12366, 12386, 12406, and 12426 plus port-offset). For more information on this refer to "Firewall Ports for Viptela Deployments".

If Device Feature Templates are being used instead of CLI template, then to achieve the same, we need to update or add new VPN Ethernet Feature Template for corresponding transport (vpn 0) interface with New Port Forwarding Rule, as shown in the image:
Explicit ACL

Also, another solution with an explicit ACL is possible. If implicit-acl-logging is configured under policy section, you may notice the following message in the /var/log/tmplog/vdebug file:

local7.notice: Jun  8 17:53:29 vEdge2 FTMD[980]: %Viptela-vEdge2-FTMD-5-NTCE-1000026: FLOW LOG
vpn-0 198.51.100.232/42346 192.168.9.233/12346 udp: tos: 192 inbound-acl, Implicit-ACL, Result: denyPkt count 2: Byte count 342 Ingress-Intf ge0/1 Egress-intf cpu

It explains the root cause and hence you need to explicitly allow incoming data plane packets in the Access Control List (ACL) on vEdge2 like this:

```plaintext
vpn 0
interface ge0/1
    ip address 192.168.9.233/24
    nat
        respond-to-ping
    !
tunnel-interface
    encapsulation ipsec
    color biz-internet
    no allow-service bgp
    no 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
    allow-service https
!
    mtu 1506
    no shutdown
    access-list DATA_PLANE in
!
!
policy
    implicit-acl-logging
    access-list DATA_PLANE
        sequence 10
            match
destination-port 12346 12445 protocol 17 ! action accept !! default-action drop !!
```

If Device Feature Templates are being used, then you need to create Localized Policy and configure ACL on Configure Access Control Lists wizard step:
If `implicit-acl-logging` is not yet enabled, it might be a good idea to enable it on the final step before click on **Save Policy** button:

Localized policy (named **LOCAL_POLICY** in our case) should be referenced in the Device Template:
And then ACL (named **DATA_PLANE** in our case) should be applied under VPN Interface Ethernet Feature Template in the ingress (in) direction:

Once ACL is configured and applied to the interface to bypass data plane traffic, BFD session is more to the **up** state again:

```
vEdge2# show tunnel statistics dest-ip 198.51.100.232 ; show bfd sessions site-id 232
```
Other Considerations

Please note that workaround with ACL is much more practical than NAT port-forwarding because you may also match based on source addresses of the remote site for greater security and to protect from DDoS attacks to your device, e.g:

```
access-list DATA_PLANE
sequence 10
match
  source-ip 198.51.100.232/32
  destination-port 12346 12445
  protocol 17
! action accept
!`

Also please note that for any other incoming traffic (not specified with allowed-services) e.g. for default iperf port 5001 explicit ACL seq 20 like in this example this won't make any effect as opposed to data plane traffic:

```
policy
access-list DATA_PLANE
sequence 10
match
  source-ip 198.51.100.232/32
  destination-port 12346 12445
  protocol 17
! action accept
!
sequence 20
match
  destination-port 5001
  protocol 6
```
And you still need NAT port-forward exemption rule for `iperf` to work:

```
vEdgeCloud2# show running-config vpn 0 interface ge0/1 nat
vpn 0
interface ge0/1
 nat
    respond-to-ping
    port-forward port-start 5001 port-end 5001 proto tcp
    private-vpn       0
    private-ip-address 192.168.9.233

!  
!

Conclusion

This is expected behavior on vEdge routers caused by NAT software design specifics and can’t be avoided.