

As conexões TCP não conseguem estabelecer quando o tráfego segue caminhos assimétricos

Contents

[Introduction](#)

[Problema](#)

[Diagrama de topologia](#)

[Diagnóstico](#)

[Solução](#)

[Conclusão](#)

Introduction

Este documento descreve o problema que surge quando os caminhos assimétricos são usados para o encaminhamento de tráfego na malha SD-WAN.

Problema

As conexões Secure Shell (SSH) não podem ser estabelecidas para host2 (nome de host - edgeclient2) do host1 (nome de host - edgeclient1), mas ao mesmo tempo o SSH funciona bem na direção inversa.

```
[root@edgeclient2 user]# ssh user@192.168.40.21
user@192.168.40.21's password:
Last login: Sun Feb 10 13:26:32 2019 from 192.168.60.20
[user@edgeclient1 ~]$
```

```
[root@edgeclient1 user]# ssh user@192.168.60.20
<nothing happens after that>
```

or

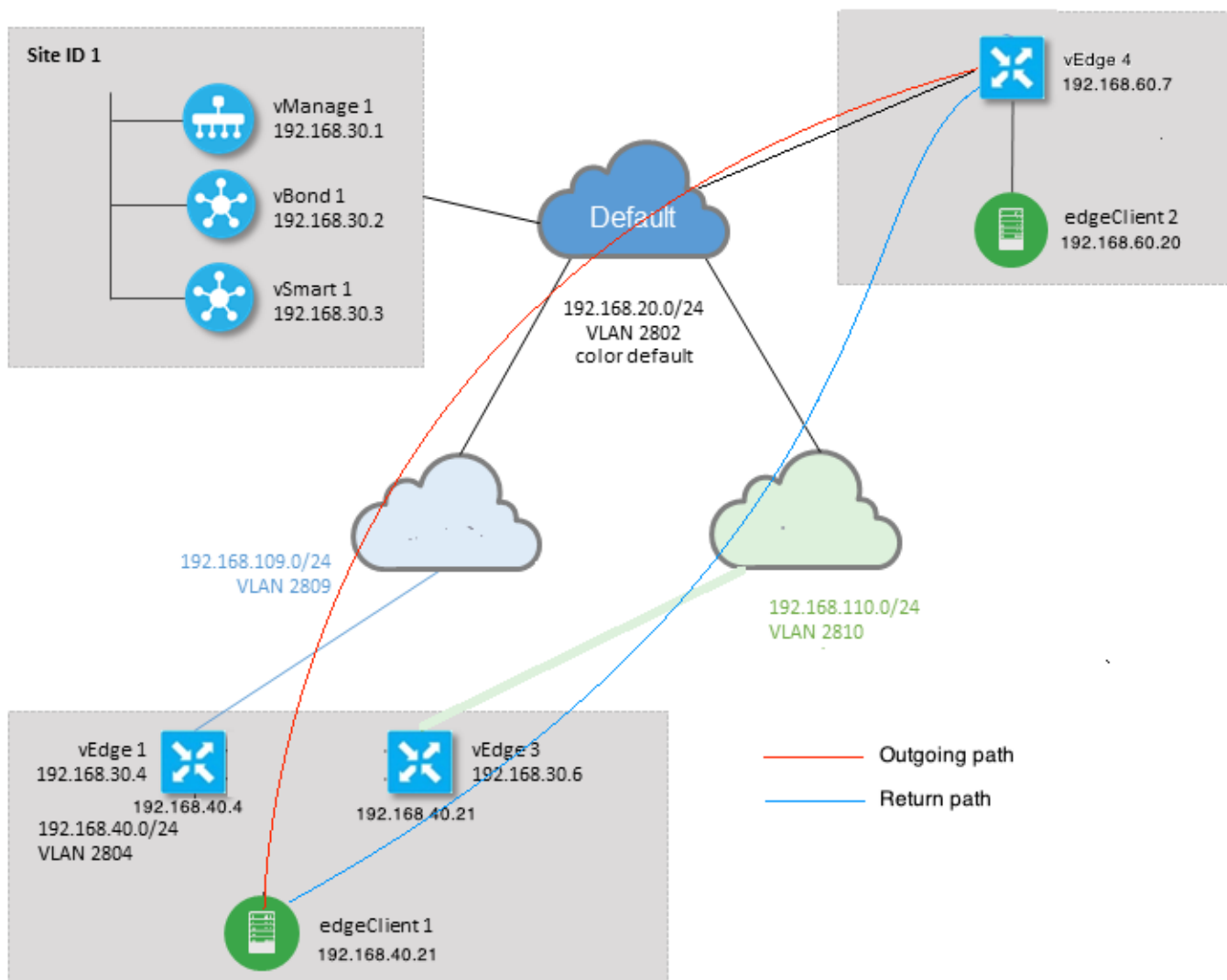
```
[user@edgeclient1 ~]$ ssh user@192.168.60.20
ssh_exchange_identification: Connection closed by remote host
```

Os daemons e os clientes SSH do cliente1 e do cliente2 têm configurações boas conhecidas e as conexões podem ser estabelecidas a partir do segmento de LAN local com êxito:

```
vedge4# request execute vpn 40 ssh user@192.168.60.20
user@192.168.60.20's password:
Last login: Sun Feb 10 13:28:23 2019 from 192.168.60.7
[user@edgeclient2 ~]$
```

Todas as outras aplicações do Transmission Control Protocol (TCP) têm problemas semelhantes.

Diagrama de topologia



Diagnóstico

Essas listas de controle de acesso (ACLs) foram configuradas e aplicadas nas direções correspondentes nas interfaces do lado de serviço do vEdge1 e vEdge3:

```
policy
access-list SSH_IN
sequence 10
match
source-ip      192.168.40.21/32
destination-ip 192.168.60.20/32
!
action accept
count SSH_IN
!
!
default-action accept
!
access-list SSH_OUT
sequence 10
match
```

```

source-ip      192.168.60.20/32
destination-ip 192.168.40.21/32
!
action accept
count SSH_OUT
!
!
default-action accept
!
!

```

A ACL espelhada foi aplicada no vEdge4:

```

policy
access-list SSH_IN
sequence 10
match
source-ip      192.168.60.20/32
destination-ip 192.168.40.21/32
!
action accept
count SSH_IN
!
!
default-action accept
!
access-list SSH_OUT
sequence 10
match
source-ip      192.168.40.21/32
destination-ip 192.168.60.20/32
!
action accept
count SSH_OUT
!
!
default-action accept
!
!

```

Além disso, a visibilidade do aplicativo foi ativada em todos os roteadores vEdge e os fluxos foram verificados durante a fase de estabelecimento da conexão SSH:

```
vedgel# show app cflowd flows | tab ; show policy access-list-counters
```

| TIME | EGRESS | | INGRESS | TCP | | | | | | | TOTAL | | |
|-------|---------------|---------------|--------------------------|------|------|-------|-------|-------|---------------|--------|-------|----|------|
| | MIN | MAX | | SRC | DEST | IP | CNTRL | ICMP | TOTAL | | | | |
| TOTAL | SRC | IP | DEST | IP | PORT | PORT | DSCP | PROTO | BITS | OPCODE | NHOP | IP | PKTS |
| BYTES | LEN | LEN | START | TIME | PORT | PORT | EXP | NAME | NAME | | | | |
| 40 | 192.168.40.21 | 192.168.60.20 | 47866 | 22 | 0 | 6 | 24 | 0 | 192.168.109.7 | 3 | | | |
| 227 | 66 | 87 | Sun Feb 17 14:13:25 2019 | 34 | | ge0/0 | ge0/1 | | | | | | |

```

COUNTER
NAME      NAME      PACKETS  BYTES

```

```
-----
SSH_IN  SSH_IN  3      227
SSH_OUT SSH_OUT  2      140
```

```
vedge3# show app cflowd flows | tab ; show policy access-list-counters
```

```

                                     TCP
TIME      EGRESS  INGRESS
TOTAL    MIN    MAX
VPN      SRC IP      DEST IP      SRC  DEST      IP      CNTRL  ICMP
BYTES   LEN    LEN  START TIME  PORT  PORT    DSCP  PROTO  BITS  OPCODE  NHOP IP      PKTS
-----
40      192.168.60.20  192.168.40.21  22   47866  0      6      18    0      192.168.40.21  8
480     60    60   Sun Feb 17 14:14:08 2019  51      ge0/1  ge0/0
```

```

COUNTER
NAME     NAME     PACKETS  BYTES
-----
SSH_IN  SSH_IN   0        0
SSH_OUT SSH_OUT  7        420
```

```
vedge4# show app cflowd flows | tab ; show policy access-list-counters
```

```

                                     TCP
TIME      EGRESS  INGRESS
TOTAL    TOTAL  MIN    MAX
VPN      SRC IP      DEST IP      SRC  DEST      IP      CNTRL  ICMP
BYTES   LEN    LEN  START TIME  PORT  PORT    DSCP  PROTO  BITS  OPCODE  NHOP IP      PKTS
-----
40      192.168.40.21  192.168.60.20  47866  22    0      6      2     0      192.168.60.20  4
240     60    60   Sun Feb 17 14:17:44 2019  37      ge0/2  ge0/0
40      192.168.60.20  192.168.40.21  22     47866  0      6      18    0      192.168.110.6  8
592     74    74   Sun Feb 17 14:17:44 2019  49      ge0/0  ge0/2
```

```

COUNTER
NAME     NAME     PACKETS  BYTES
-----
SSH_IN  SSH_IN   8        592
SSH_OUT SSH_OUT  4        240
```

Como você pode ver nessas saídas, os fluxos de entrada e saída são assimétricos. o edgeclient1 (192.168.40.21) está tentando estabelecer uma sessão SSH com o edgeclient2 (192.168.60.20) e o tráfego de entrada vem via vEdge1 e retorna as devoluções de tráfego via vEdge3. Nos contadores da ACL, você também pode ver que o número de pacotes de entrada e saída no vEdge4 não corresponde à soma nas direções correspondentes no vEdge1 e vEdge3. Ao mesmo tempo, não há perda de pacote ao testar com o ping:

```
[root@edgeclient1 user]# ping -f 192.168.60.20 -c 10000
PING 192.168.60.20 (192.168.60.20) 56(84) bytes of data.
```

```
--- 192.168.60.20 ping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 3076ms
rtt min/avg/max/mdev = 0.128/0.291/6.607/0.623 ms, ipg/ewma 0.307/0.170 ms
```

```
[root@edgeclient2 user]# ping -f 192.168.40.21 -c 10000
PING 192.168.40.21 (192.168.40.21) 56(84) bytes of data.
```

```
--- 192.168.40.21 ping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 3402ms
rtt min/avg/max/mdev = 0.212/0.318/2.766/0.136 ms, ipg/ewma 0.340/0.327 ms
```

Também recapitula que o SSH funciona bem na direção inversa e os arquivos podem ser copiados por scp/sftp também sem nenhum problema.

Solução

Inicialmente, suspeitava-se de alguma configuração ou política de dados de inspeção profunda de pacotes (DPI), mas nenhuma delas foi ativada:

```
vedge3# show policy from-vsmart
% No entries found.
```

```
vedge1# show policy from-vsmart
% No entries found.
```

No entanto, descobriu-se que a otimização do TCP estava ativada:

```
vedge1# show app tcp-opt active-flows
```

| RX | UNOPT | PROXY | SRC | DEST | EGRESS | INGRESS | TX | |
|-------|---------------|---------------|--------------|------|--------------------------|---------|-------|-------|
| VPN | SRC IP | DEST IP | PORT | PORT | START TIME | NAME | NAME | BYTES |
| BYTES | TCP STATE | REASON | IDENTITY | | | | | |
| 40 | 192.168.40.21 | 192.168.60.20 | 47868 | 22 | Sun Feb 17 14:18:13 2019 | ge0_0 | ge0_1 | 314 |
| 0 | In-progress | - | Client-Proxy | | | | | |

```
vedge1# show app tcp-opt expired-flows
```

| TX | RX | UNOPT | PROXY | SRC | DEST | START TIME | END |
|----------------------|-----|---------------|---------------|-----------|--------|--------------------------|---------------|
| TIMESTAMP | VPN | SRC IP | DEST IP | PORT | PORT | DELETE REASON | |
| TIME | | BYTES | BYTES | TCP STATE | REASON | IDENTITY | |
| 1549819969608 | 40 | 192.168.40.21 | 192.168.60.7 | 22 | 56612 | Sun Feb 10 18:32:49 2019 | Sun |
| Feb 10 18:36:03 2019 | | 5649 | 4405 | Optimized | - | Server-Proxy | CLOSED |
| 1549820055487 | 40 | 192.168.40.21 | 192.168.60.7 | 22 | 56613 | Sun Feb 10 18:34:15 2019 | Sun |
| Feb 10 19:07:46 2019 | | 5719 | 4669 | Optimized | - | Server-Proxy | CLOSED |
| 1550408210511 | 40 | 192.168.40.21 | 192.168.60.20 | 47862 | 22 | Sun Feb 17 13:56:50 2019 | Sun |
| Feb 17 13:56:58 2019 | | 401 | 0 | Optimized | - | Client-Proxy | STATE-TIMEOUT |
| 1550408981634 | 40 | 192.168.40.21 | 192.168.60.20 | 47864 | 22 | Sun Feb 17 14:09:41 2019 | Sun |
| Feb 17 14:09:49 2019 | | 401 | 0 | Optimized | - | Client-Proxy | STATE-TIMEOUT |
| 1550409205399 | 40 | 192.168.40.21 | 192.168.60.20 | 47866 | 22 | Sun Feb 17 14:13:25 2019 | Sun |
| Feb 17 14:13:33 2019 | | 227 | 0 | Optimized | - | Client-Proxy | STATE-TIMEOUT |
| 1550409493042 | 40 | 192.168.40.21 | 192.168.60.20 | 47868 | 22 | Sun Feb 17 14:18:13 2019 | Sun |
| Feb 17 14:18:21 2019 | | 401 | 0 | Optimized | - | Client-Proxy | STATE-TIMEOUT |

Além disso, em **debugs ftcpopt** **CONN_TEARDOWN** pode ser vista uma mensagem.

```

vedge1# show log /var/log/tmplog/vdebug tail "-f"
local7.debug: Feb 17 13:56:50 vedge1 FTMD[662]: ftm_tcptopt_flow_add[268]: Created new tcpflow :-
vrid-3 192.168.40.21/47862 192.168.60.20/22
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[388]: Trying to
pack and send the following message to TCPD
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[408]: Sending
following CONN_TD msg
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[413]:
192.168.40.21:47862->192.168.60.20:22; vpn:40; syn_seq_num:4172167164; identity:0; cport_prime:0
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_msgq_tx[354]: Transferring size = 66
bytes data
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[416]: Successfully
sent conn_td msg to TCPD
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_propagate_tear_down[1038]: Sent
CONN_TEARDOWN msg to tcpd for existing tcpflow :- vrid-3 192.168.40.21/47862 192.168.60.20/22 ;
identity:CLIENT_SIDE_PROXY . Send Successful !
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_append_expired_err_flow_tbl[958]:
Appending flow vrid-3 192.168.40.21/47862 192.168.60.20/22 to the expired flow table at Sun Feb
17 13:56:58 2019
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_append_expired_err_flow_tbl[980]:
Appending flow vrid-3 192.168.40.21/47862 192.168.60.20/22 to the error flow table at Sun Feb
17 13:56:58 2019
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_flow_delete[293]: Removing tcpflow :-
vrid-3 192.168.40.21/47862 192.168.60.20/22
local7.debug: Feb 17 13:56:58 vedge1 TCPD[670]: handle_upstream_connect[538]: Error - BP NULL
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_msg_decode[254]: FTM-TCPD: Received
FTM_TCPD__PB_FTM_TCPD_MSG__E_MSG_TYPE__CONN_CLOSED msg
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[139]: FTM-TCPD:
Received CONN_CLOSED for following C->S
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[150]:
192.168.40.21:47862->192.168.60.20:22; vpn:40; syn_seq_num:4172167164; identity:0;
cport_prime:47862; bind_port:0
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[184]: FTM-TCPD:
Could not find entry in FT for following flow
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[185]: vrid-3
192.168.40.21/47862 192.168.60.20/22

```

E aqui você pode ver um exemplo quando a otimização do TCP funciona corretamente (mensagem CONN_EST pode ser vista):

```

vedge3# show log /var/log/tmplog/vdebug tail "-f -n 0"
local7.debug: Feb 17 15:41:13 vedge3 FTMD[657]: ftm_tcpd_msg_decode[254]: FTM-TCPD: Received
FTM_TCPD__PB_FTM_TCPD_MSG__E_MSG_TYPE__CONN_CLOSED msg
local7.debug: Feb 17 15:41:13 vedge3 FTMD[657]: ftm_tcpd_handle_conn_closed[139]: FTM-TCPD:
Received CONN_CLOSED for following C->S
local7.debug: Feb 17 15:41:13 vedge3 FTMD[657]: ftm_tcpd_handle_conn_closed[150]:
192.168.40.21:47876->192.168.60.20:22; vpn:40; syn_seq_num:2779178897; identity:0;
cport_prime:47876; bind_port:0
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcpd_msg_decode[258]: FTM-TCPD: Received
FTM_TCPD__PB_FTM_TCPD_MSG__E_MSG_TYPE__CONN_EST msg
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcpd_handle_conn_est[202]: FTM-TCPD:
Received CONN_EST for following C->S
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcpd_handle_conn_est[213]:
192.168.40.21:47878->192.168.60.20:22; vpn:40; syn_seq_num:2690847868; identity:0;
cport_prime:47878; bind_port:0
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcptopt_flow_add[268]: Created new tcpflow :-
vrid-3 192.168.40.21/47878 192.168.60.20/22

```

Conclusão

A otimização de TCP exige que os fluxos sejam simétricos, portanto, para resolver esse problema, a otimização de TCP deve ser desabilitada (**sem otimização de tcp de vpn 40**) ou a **política de dados deve ser criada para forçar os fluxos de TCP a seguirem o mesmo caminho em ambas as direções**. Você pode encontrar mais informações sobre isso na seção [SD-WAN Design Guide](#) Traffic Symmetry for DPI, página 23.