트래픽이 비대칭 경로를 따를 때 TCP 연결이 설정되지 않음

목차

소개

이 문서에서는 비대칭 경로가 SD-WAN 패브릭에서 트래픽 전달에 사용될 때 발생하는 문제를 설명합니다.

문제

SSH(Secure Shell) 연결은 host1(hostname - edgeline1)에서 host2(hostname - edgeline2)에 설정할 수 없지만, 동시에 SSH는 반대 방향으로 원활하게 작동합니다.

```
[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>
```

또는

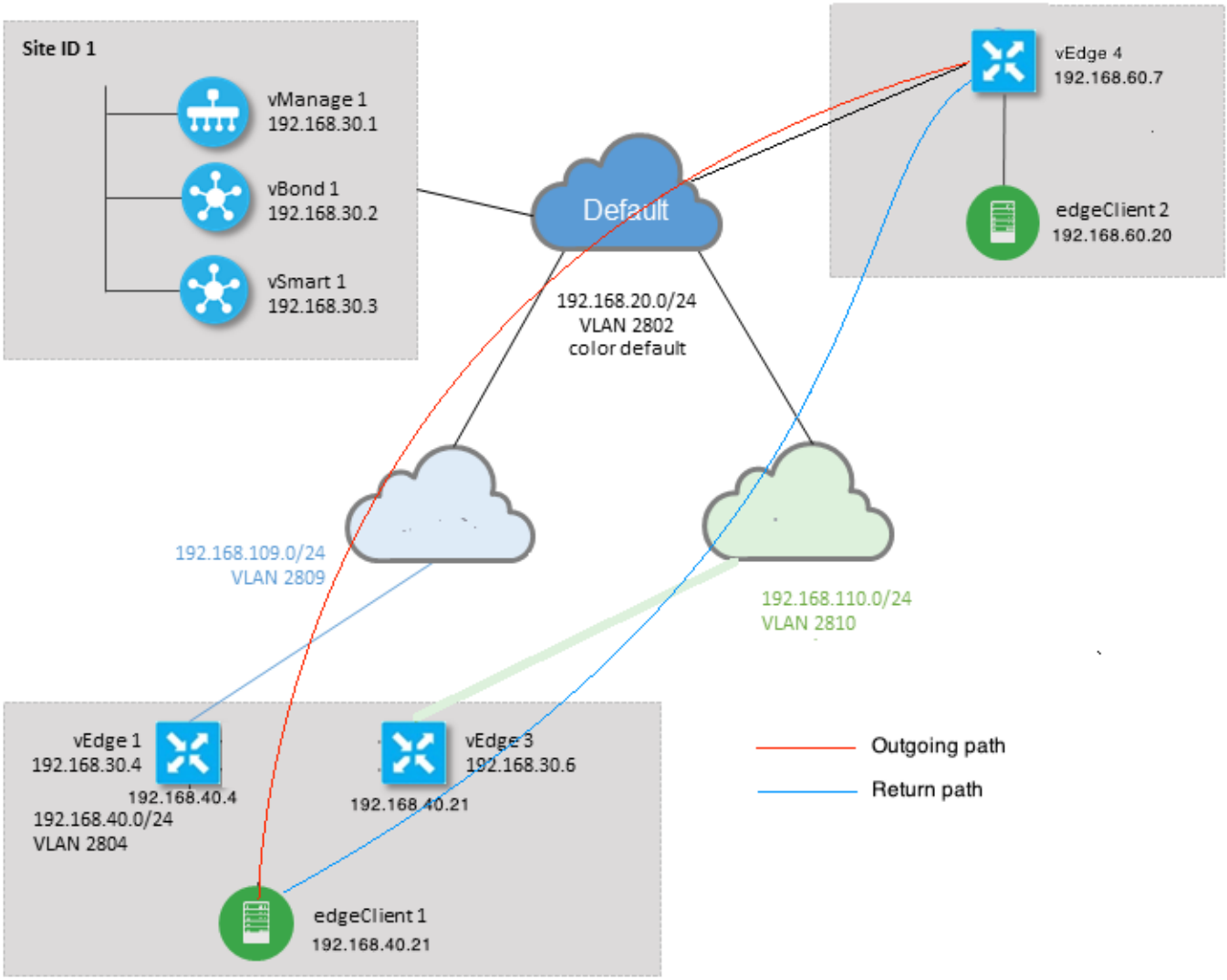
```
[user@edgeclient1 ~]$ ssh user@192.168.60.20
ssh_exchange_identification: Connection closed by remote host
```

edgeclient1 및 edgeclient2 SSH 데몬과 클라이언트는 모두 정상 구성 및 로컬 LAN 세그먼트에서 연결을 성공적으로 설정할 수 있습니다.

```
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 ~]$
```

다른 모든 TCP(Transmission Control Protocol) 애플리케이션도 비슷한 문제를 안고 있습니다.

토폴로지 다이어그램



진단

이 ACL(Access Control List)은 vEdge1 및 vEdge3의 서비스 측 인터페이스에서 해당 방향으로 구성 및 적용되었습니다.

```
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
 !
!
```

vEdge4에 미러 ACL이 적용되었습니다.

```
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
 !
!
```

또한 모든 vEdge 라우터에서 앱 가시성이 활성화되었으며 SSH 연결 설정 단계에서 플로우가 검사되었습니다.

```
vedge1# show app cflowd flows | tab ; show policy access-list-counters

                                                                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
-------------------------------------------------------------------------------------------------
-----------------------------------------------------------------
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
                                    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
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------
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
                                    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
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------
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
```

이러한 출력에서 볼 수 있듯이 인바운드 및 아웃바운드 흐름은 비대칭적입니다 .edgelient1(192.168.40.21)이 edgelient2(192.168.60.20)를 사용하여 SSH 세션을 설정하려고 시도하고 수신 트래픽은 vEdge1을 통해 오고 vEdge3를 통해 트래픽 반환 트래픽을 반환합니다. ACL 카운터에서 vEdge4의 수신 및 발신 패킷 수가 vEdge1 및 vEdge3의 해당 방향에서 합과 일치하지 않음을 확인할 수 있습니다. 동시에 ping으로 테스트할 때 패킷 손실이 발생하지 않습니다.

```
[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
```

또한 SSH는 반대 방향으로 잘 작동하며 scp/sftp를 통해 문제 없이 파일을 복사할 수 있음을 다시 매핑합니다.

솔루션

일부 DPI(Deep Packet Inspection) 컨피그레이션 또는 데이터 정책이 초기에 의심되었지만 활성화되지 않았습니다.

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

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

그러나 결국 TCP 최적화가 활성화되었다는 사실이 확인되었습니다.

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

                                                                                    EGRESS  INGRESS
                                    SRC    DEST                                     INTF    INTF     TX
RX                  UNOPT   PROXY
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

                                              SRC    DEST
TX     RX                UNOPT   PROXY
TIMESTAMP      VPN  SRC IP         DEST IP        PORT   PORT   START TIME                END
TIME                   BYTES  BYTES  TCP STATE  REASON  IDENTITY      DELETE REASON
-------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------
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
```

또한 ftm tcpopt **CONN**_TEARDOWN 메시지도 볼 수 있습니다.

```
vedge1# show log /var/log/tmplog/vdebug tail "-f"
```

```
local7.debug: Feb 17 13:56:50 vedge1 FTMD[662]: ftm_tcpopt_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]: Transfering 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_tcpopt_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_tcpopt_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_tcpopt_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_tcpopt_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
```

TCP 최적화가 제대로 작동하는 경우의 예를 여기서 확인할 수 있습니다(CONN_EST 메시지를 볼 수 있음).

```
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_tcpopt_flow_add[268]: Created new tcpflow :-
vrid-3 192.168.40.21/47878 192.168.60.20/22
```

결론

TCP 최적화를 위해서는 흐름이 대칭이어야 하므로 이 문제를 해결하려면 TCP 최적화를 비활성화해야 합니다(**vpn 40 tcp 최적화 안 함). 또는 TCP 플로우가 양방향으로 동일한 경로를 사용하도록 하려면 데이터 정책을 만들어야 합니다.** 이에 대한 자세한 내용은 SD-WAN 설계 가이드 섹션 DPI의 트래픽 대칭(Traffic Symmetry for DPI)23페이지에서 확인할 수 있습니다.