Impossibile stabilire le connessioni TCP quando il traffico segue percorsi asimmetrici

Sommario

Introduzione

In questo documento viene descritto il problema che si verifica quando si utilizzano percorsi asimmetrici per l'inoltro del traffico nell'infrastruttura SD-WAN.

Problema

Non è possibile stabilire connessioni Secure Shell (SSH) all'host 2 (hostname - edgeclien2) dall'host 1 (hostname - edgeclien1), ma allo stesso tempo SSH funziona correttamente in direzione 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>
```

o

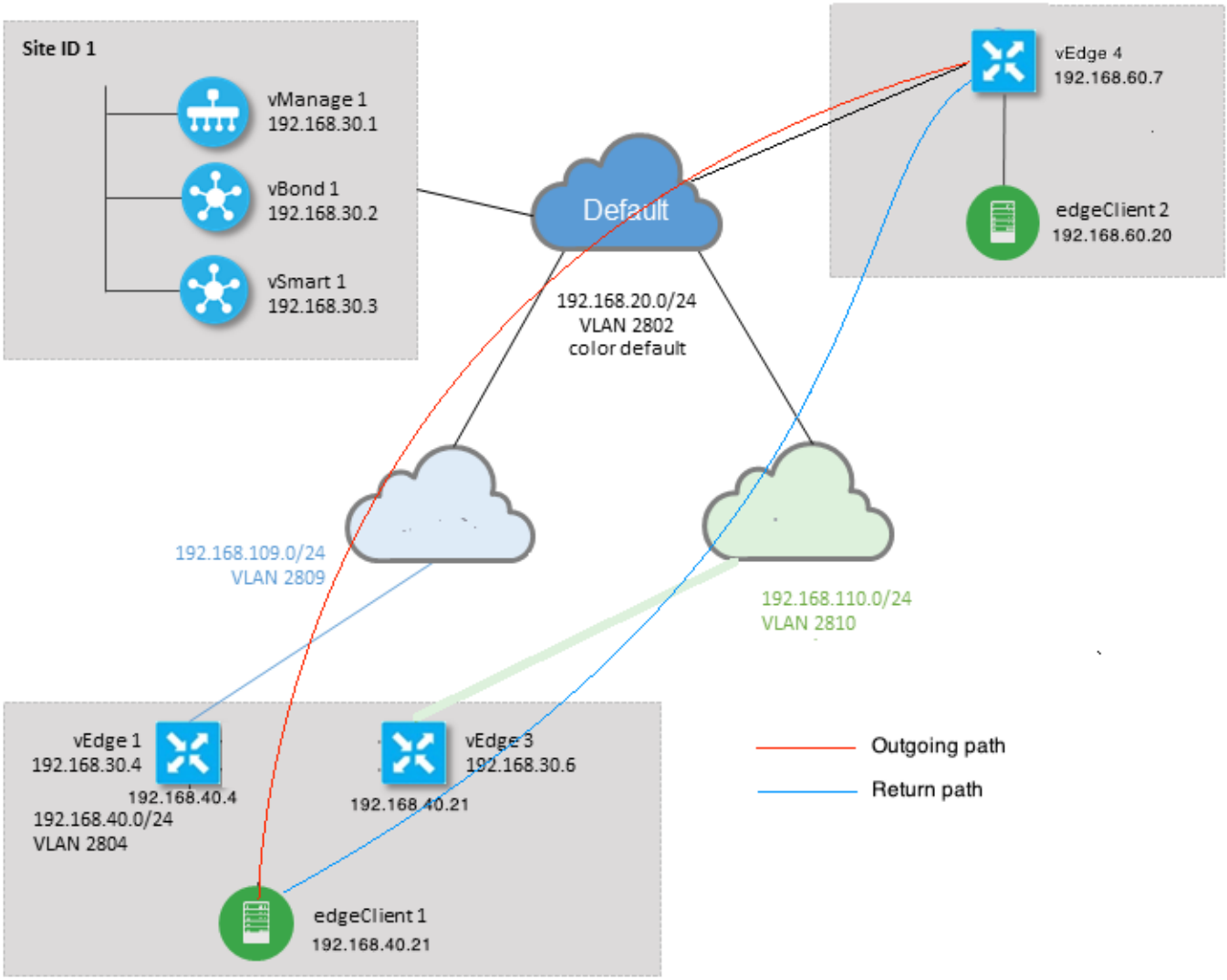
```
[user@edgeclient1 ~]$ ssh user@192.168.60.20
ssh_exchange_identification: Connection closed by remote host
```

Sia i daemon SSH edgeclient1 che edgeclient2 dispongono di configurazioni e connessioni valide che possono essere stabilite dal segmento LAN locale correttamente:

```
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 ~]$
```

Tutte le altre applicazioni TCP (Transmission Control Protocol) hanno problemi simili.

Diagramma topologico



Diagnostica

Questi elenchi di controllo di accesso (ACL) sono stati configurati e applicati nelle direzioni corrispondenti sulle interfacce del lato servizio di 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
 !
!
```

ACL con mirroring applicato a 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
 !
!
```

Anche la visibilità delle app è stata abilitata su tutti i router vEdge e i flussi sono stati controllati durante la fase di creazione della connessione SSH:

```
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
```

Come si può vedere da questi output, i flussi in entrata e in uscita sono asimmetrici. edgeclient1 (192.168.40.21) sta tentando di stabilire una sessione SSH con edgeclient2 (192.168.60.20), il traffico in entrata via vEdge1 e il traffico di ritorno via vEdge3. Dai contatori ACL è possibile verificare che il numero di pacchetti in entrata e in uscita su vEdge4 non corrisponde alla somma nelle direzioni corrispondenti su vEdge1 e vEdge3. Allo stesso tempo, non si verifica alcuna perdita di pacchetti durante il test con **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
```

Sappiamo anche che SSH funziona bene in direzione inversa e i file possono essere copiati su SCP/SFTP anche senza problemi.

Soluzione

Inizialmente si sospettava che fossero presenti una configurazione DPI (Deep Packet Inspection) o criteri dati, ma nessuno di essi è stato attivato:

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

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

Ma alla fine è stato scoperto che l'ottimizzazione TCP era abilitata:

```
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
```

Inoltre, nei **debug ftm** può essere visualizzato il messaggio **tcpopt** CONN_TEARDOWN.

```
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
```

A questo punto è possibile osservare un esempio di funzionamento corretto dell'ottimizzazione TCP (è possibile visualizzare il messaggio CONN_EST):

```
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
```

Conclusioni

L'ottimizzazione TCP richiede che i flussi siano simmetrici, quindi per risolvere il problema è necessario disabilitare l'ottimizzazione TCP (**senza ottimizzazione tcp vpn 40) o creare criteri dati per forzare i flussi TCP a seguire lo stesso percorso in entrambe le direzioni.** Per ulteriori informazioni su questo argomento, consultare la sezione Traffic Symmetry for DPI della SD-WAN Design Guide, pagina 23.