Troubleshoot FTD Cluster Asymmetry Causing TCP Connection Failures

Contents

Issue

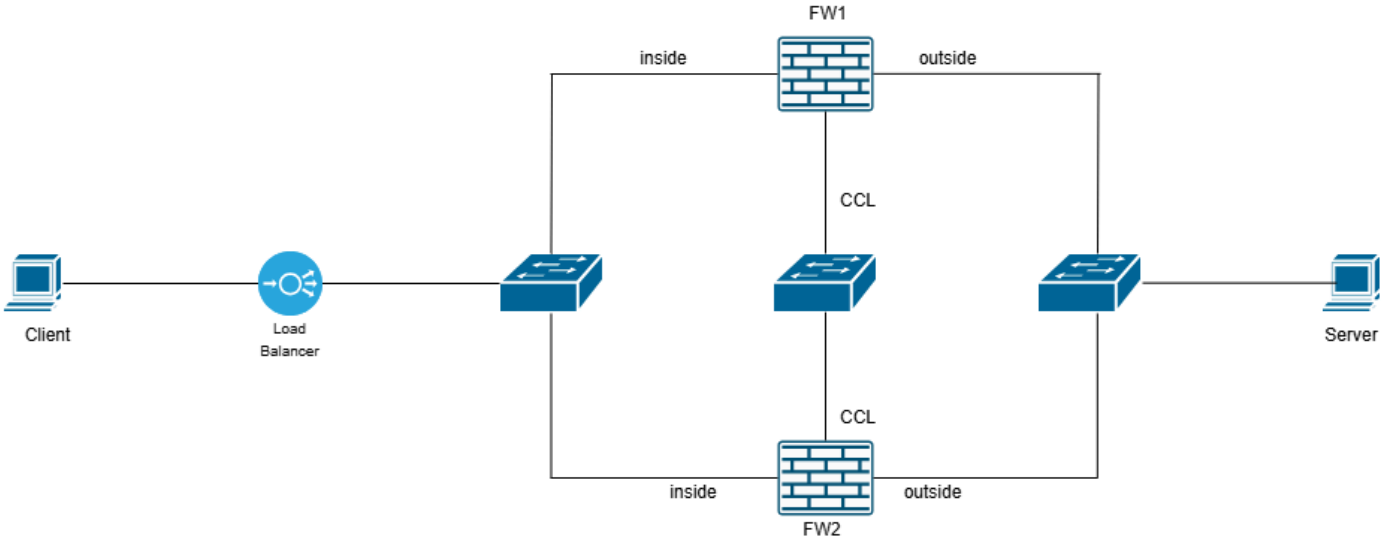
One or more of these symptoms can appear:

- Intermittent connectivity failures for applications traversing an FTD cluster
- TCP three-way handshake fails during connection attempts.
- Client sends a SYN packet, but does not receive the expected SYN-ACK response.
- Client sends a RST packet after the initial SYN.

Environment

- First seen in Secure Firewall Threat Defense 7.4 — other versions can be also affected
- Cluster configuration
- Load balancer in the network path — this is optional

Topology



inline_image_0.png

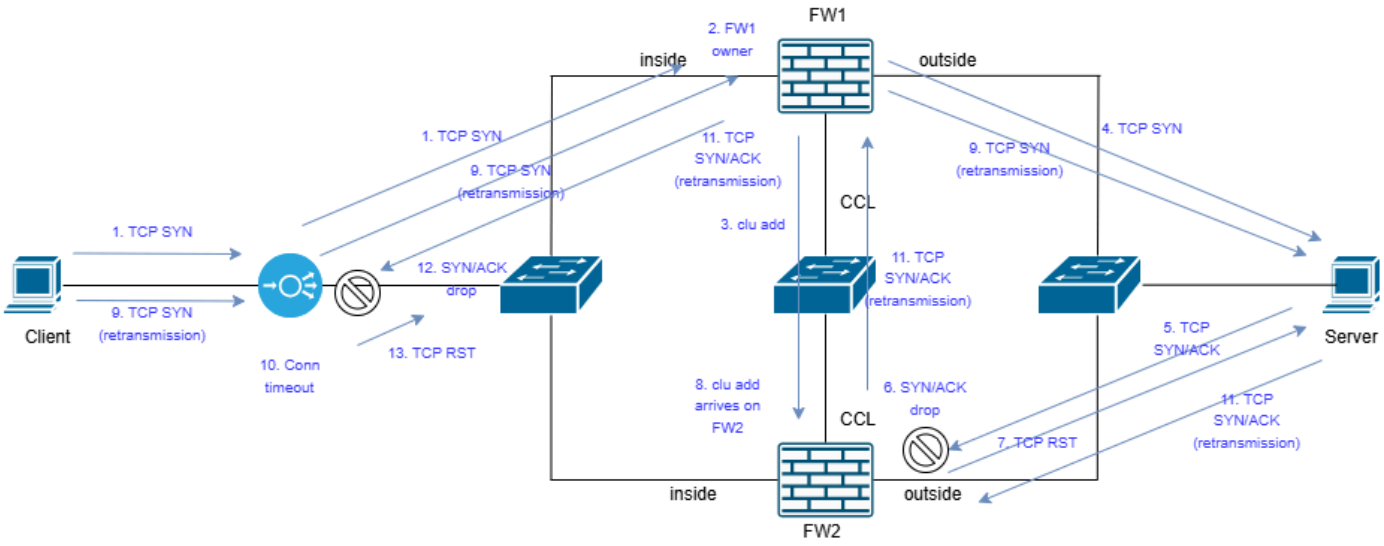
Resolution

To root-cause the problem, you need to take simultaneous captures at these points:

- FW1 inside interface (with reinject-hide)
- FW1 outside interface (with reinject-hide)
- FW1 cluster interface (CCL)
- FW2 inside interface (with reinject-hide)
- FW2 outside interface (with reinject-hide)
- FW2 cluster interface (CCL)
- Client (or as close to the client as possible)
- Server (or as close to the server as possible)

For details on how to configure the captures check: [How to Enable the Cluster Captures.]

Captures taken on both firewalls along with client and server reveal this topology:



inline_image_0.png

1. Client sends TCP SYN. The packet arrives to the load balancer (LB) and is sent to FW1.

2. FW1 receives the TCP SYN packet and becomes the flow owner.

3. FW1 informs the director (FW2) about the flow owner by sending a special (**clu add**) cluster message.

4. FW1 forwards the TCP SYN to the destination server.

Note: Steps 3 and 4 happen in no specific order.

---

5. The server replies with SYN/ACK. In this case, we have an asymmetric flow since the SYN/ACK is sent towards channel load-balancing algorithm.

6. SYN/ACK arrives on FW2 before the clu add message. This is a race condition and is purely environmental (such

7. A TCP RST is sent to the server.

8. The clu add message arrives on FW2.

9. The Client retransmits the TCP SYN packet. The TCP SYN packet is forwarded to the destination server.

10. On the LB, the TCP connection for the specific flow times out.

11. The server replies with SYN/ACK (TCP retransmission). The SYN/ACK packet arrives on FW2. This time, FW **clu add** message and the SYN/ACK is forwarded to the flow owner over the CCL. The SYN/ACK is sent to the clie

12. The LB does not know about this flow and drops the SYN/ACK. Thus, the SYN/ACK never arrives on the client

13. The LB one or more TCP RST packets.

Firewall Capture with Trace Analysis

In these outputs, captures were collected from the firewall on CCL and server-facing interfaces:

• On CCL the capture is on UDP 4193 port.

• On the data interfaces the capture matches TCP traffic between the endpoints using the **reinject-hide** option. The reason is that we want to see where the packets actually arrive.

• IP address 192.0.2.65 = client

• IP address 192.0.2.6 = server

Step 1: Use this command on the firewall device that gets the SYN/ACK to see when the **clu add** message arrived. I

<#root>

```
firepower#
```

show capture CCL decode

```
3 packets captured
   1: 08:14:20.630521       127.2.1.1.51475 > 127.2.2.1.4193:  udp 820
        Cluster ASP message: sender: 1, receiver: 0
        Add flow: owner 1, director 0, backup 0,
                  ifc_in INSIDE(7020a7), ifc_out INSIDE(7020a7)
                  TCP src 192.0.2.65/37468, dest 192.0.2.6/80
```

Step 2: Trace the SYN/ACK packet and focus on the timestamp and the trace result:

<#root>

```
firepower#
```

show capture CAPI packet-number 1 trace

```
13 packets captured
   1: 08:14:20.628690       802.1Q vlan#200 P0 192.0.2.6.80 > 192.0.2.65.37468: S 2524735158:2524735158
Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Elapsed time: 1708 ns
Config:
Additional Information:
MAC Access list

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Elapsed time: 1708 ns
Config:
Implicit Rule
Additional Information:
MAC Access list

Phase: 3
Type: INPUT-ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Elapsed time: 13664 ns
Config:
Additional Information:
Found next-hop 192.168.200.140 using egress ifc  INSIDE(vrfid:0)

Phase: 4
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Elapsed time: 16104 ns
Config:
Additional Information:
Input interface: 'INSIDE'

Flow type: NO FLOW



I (0) am becoming owner



Phase: 5
Type: OBJECT_GROUP_SEARCH
Subtype:
Result: ALLOW
Elapsed time: 19520 ns
Config:
Additional Information:
 Source object-group match count:          0
 Source NSG match count:                   0
 Destination NSG match count:              0
 Classify table lookup count:              1
 Total lookup count:                       1
```

```
 Duplicate key pair count:                 0
 Classify table match count:               4

Phase: 6
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Elapsed time: 366 ns
Config:
access-group CSM_FW_ACL_ global
access-list CSM_FW_ACL_ advanced permit ip any any rule-id 268436480
access-list CSM_FW_ACL_ remark rule-id 268436480: ACCESS POLICY: mzafeiro_empty - Default
access-list CSM_FW_ACL_ remark rule-id 268436480: L4 RULE: DEFAULT ACTION RULE
Additional Information:
 This packet will be sent to snort for additional processing where a verdict will be reached

Phase: 7
Type: CONN-SETTINGS
Subtype:
Result: ALLOW
Elapsed time: 366 ns
Config:
class-map tcp
 match access-list tcp
policy-map global_policy
 class tcp
  set connection conn-max 0 embryonic-conn-max 0 random-sequence-number disable syn-cookie-mss 1380
service-policy global_policy global
Additional Information:

Phase: 8
Type: NAT
Subtype: per-session
Result: ALLOW
Elapsed time: 366 ns
Config:
Additional Information:

Phase: 9
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Elapsed time: 366 ns
Config:
Additional Information:

Result:
input-interface: INSIDE(vrfid:0)
input-status: up
input-line-status: up
output-interface: INSIDE(vrfid:0)
output-status: up
output-line-status: up

Action: drop


Time Taken: 54168 ns

Drop-reason: (tcp-not-syn) First TCP packet not SYN, Drop-location: frame snp_sp:7459 flow (NA)/NA
```

Key Points

• The **Add flow** message arrived at 08:14:20.630521 while the SYN/ACK ~2 msec earlier at 08:14:20.628690. This

• The SYN/ACK packet is dropped by the firewall with **tcp-not-syn** ASP reason. Notice that in Phase 4 the firewall tried to identify if there was a known flow owner but didn't find

This output shows a trace of the SYN/ACK when the firewall knows about the flow:

<#root>

```
firepower#

show capture CAPI packet-number 3 trace



13 packets captured

   3: 08:14:21.629560       802.1Q vlan#200 P0 192.0.2.6.80 > 192.0.2.65.37468: S 2540375172:2540375172
Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Elapsed time: 1708 ns
Config:
Additional Information:
MAC Access list

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Elapsed time: 1708 ns
Config:
Implicit Rule
Additional Information:
MAC Access list

Phase: 3
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Elapsed time: 3416 ns
Config:
Additional Information:
Input interface: 'INSIDE'
Flow type: STUB

I (0) have flow, valid owner (1).
```

```
Phase: 4
Type: CAPTURE
Subtype:
Result: ALLOW
Elapsed time: 7808 ns
Config:
Additional Information:
MAC Access list

Result:
input-interface: INSIDE(vrfid:0)
input-status: up
input-line-status: up
Action: allow
Time Taken: 14640 ns

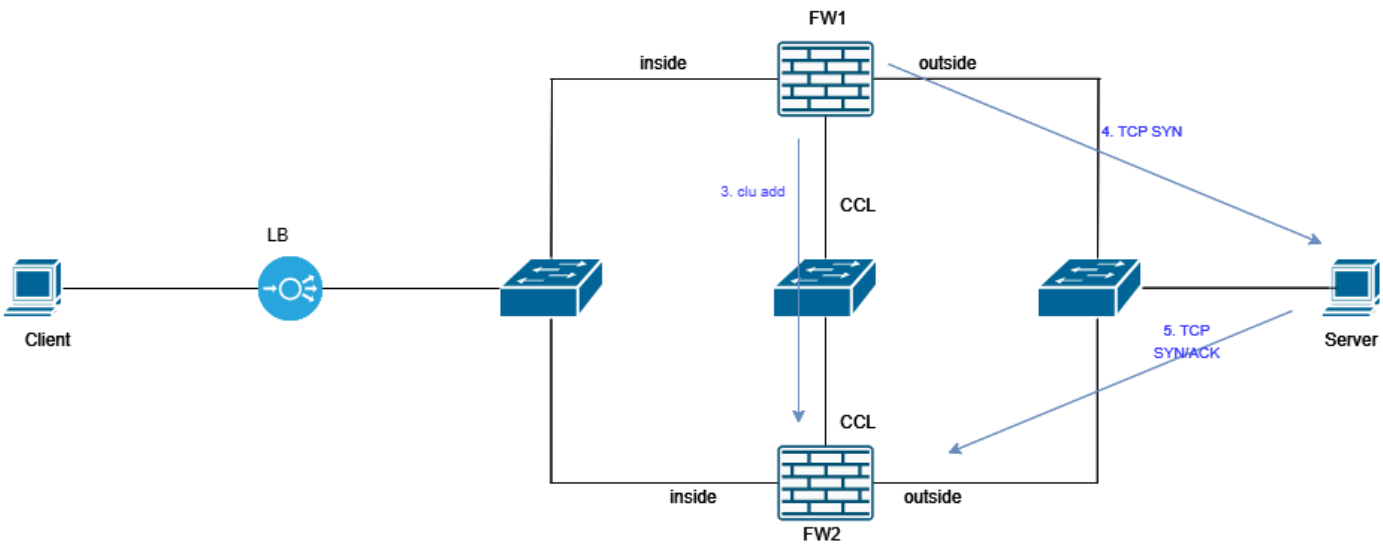
1 packet shown
firepower#
```

The key point is in Phase 3. The firewall knows that the cluster unit 1 is the flow owner. You can use the **show clust** command to see which device is unit 0 and which one is 1.

Frequently Asked Questions

Q. Why we see intermittent TCP connectivity problems?

A. Since this is a race condition, it happens randomly. The race condition can be visualized accordingly:

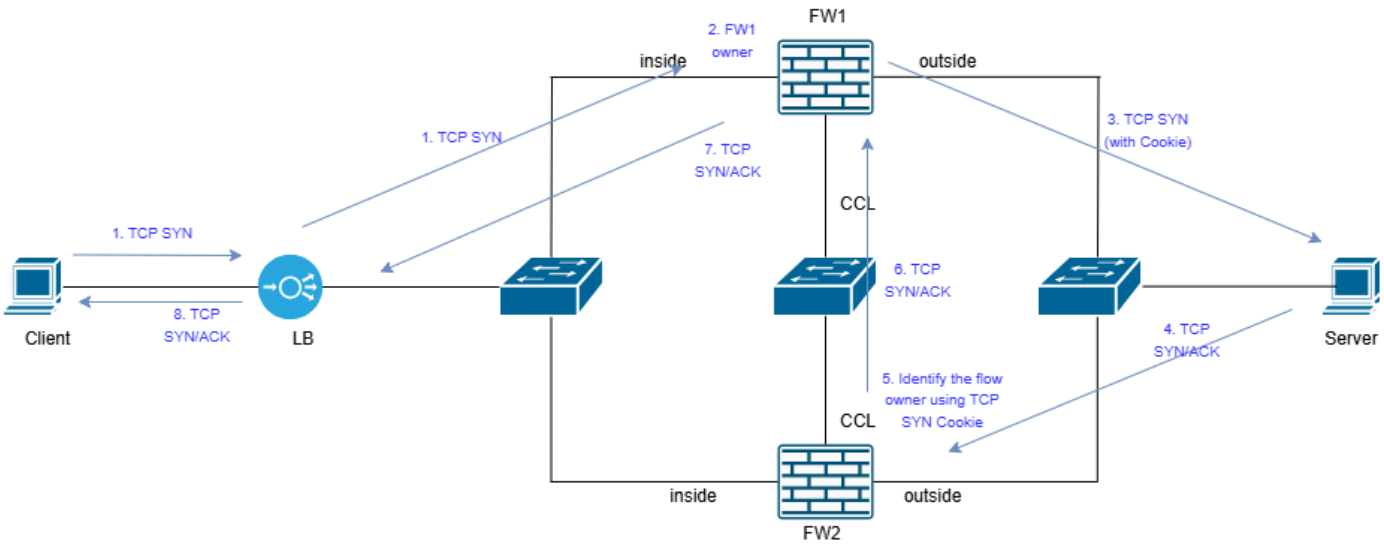


inline_image_0.png

Q. What are possible solutions to avoid the race condition?

A.

Solution 1: Enable TCP sequence number randomization to take advantage of the TCP SYN Cookie mechanism. In t



inline_image_1.png

Solution 2: Eliminate the asymmetry in the network. First, you need to identify the reason of the asymmetry. This ca channel load-balancing algorithm, rewire the port-channel cables in different order, among other things.

Cause

The root cause is a race condition caused due to cluster asymmetry within the FTD cluster deployment. The SYN-ACK packets from the server are being processed by a different FTD cluster node than the one that handled the initia

Related Content

- Cisco Technical Support & Downloads