Troubleshoot Traffic Issues on CSF CLI

Contents

Introduction

This document describes how to use Cisco Secure Firewall CLI to troubleshoot traffic issues by validating routing, packet flow, and Snort behavior.

Prerequisites

There are no specific requirements for this document.

Requirements

Cisco recommends that you have knowledge of these topics:

- Firepower Management Center (FMC) policies and object configuration
- Cisco Secure Firewall (CSF) routing and interface logic
- Packet inspection and firewall troubleshooting concepts

Components Used

This document is not restricted to specific software and hardware versions for CSF. The device used in the document is running code 7.6.5.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Related Products

This document can also be used with these hardware and software versions:

- Secure Firewall Management Center (FMC)
- Cisco Secure Firewall (CSF)
- Firepower Threat Defense CLI tools

Important Information

When a device is experiencing elevated CPU, memory, or I/O utilization, executing troubleshooting commands can exacerbate performance degradation. It is recommended to investigate and resolve the resource exhaustion condition first, then proceed with traffic issue diagnosis. Diagnostic commands consume additional system resources, which can further degrade availability and prolong recovery time.

Background Information

Traffic troubleshooting on CSF often begins by confirming the logical path that a packet must take through the firewall. Once the route and interface mapping are understood, the next step is to compare the expected path to the actual packet flow. This can be performed using route lookups, packet-tracer simulation, packet captures, and Snort engine tracing.

Each tool provides a different layer of visibility:

- **show route** confirms the packet's routing path and interface identity.
- **packet-tracer** simulates the traffic flow and identifies ACL, NAT, and interface decisions.
- **packet captures** inspect live traffic that traverses the firewall.
- **system support trace** exposes Snort decision-making for policy-related blocks or inspection events.

Tip: Start with routing and packet-tracer validation before capturing traffic, so you can confirm whether the expected path and access control decisions are correct before analyzing live traffic.

Problem

A traffic flow is not working as expected, and the administrator needs to determine whether the packet is taking the correct path, whether the firewall is permitting or denying the connection, and whether Snort or policy inspection is blocking the traffic.

Step 1: Find the Route

Identify the routing path and determine the logical interface names associated with the source and destination IP addresses. The logical interface names are required by most troubleshooting commands, so this must be the first step in the troubleshooting workflow.

Run the command for the source and destination IP address:

```
show route <IP>
```

Internet-destined traffic that relies on the default route requires identification of the logical egress interface. The logical interface name can be determined by executing this step:

```
show route 0.0.0.0
```

Example output:

```
FTD1# show route 192.168.0.11

Routing entry for 192.168.0.0 255.255.255.0
Known via "connected", distance 0, metric 0 (connected, via interface)
Routing Descriptor Blocks:

    directly connected, via Inside
    Route metric is 0, traffic share count is 1
```

```
FTD1# show route 0.0.0.0

Routing entry for 0.0.0.0 0.0.0.0, supernet
Known via "static", distance 1, metric 0, candidate default path
Routing Descriptor Blocks:

    128.107.0.1, via Outside
    Route metric is 0, traffic share count is 1
```

In this example, the logical ingress interface is Inside and the egress interface is Outside.

Caution: Network Address Translation (NAT) can redirect traffic through different interfaces than the routing table indicates when NAT policies match the traffic flow. When troubleshooting connectivity, verify that your NAT configuration aligns with the expected traffic path.

Tip: The logical interface names are used in CLI troubleshooting commands. Please take note of the logical name for future reference.

Step 2: Run Packet-Tracer

Use packet-tracer to simulate how the firewall evaluates a specific traffic flow. This tool helps confirm which interfaces are used, whether a NAT policy is applied, and whether an ACL permits or denies the traffic.

Use this command syntax:

```
packet-tracer input <source interface> <protocol> <source IP> <source port> <destination IP> <destinati
```

Example command:

```
FTD1# packet-tracer input Inside tcp 192.168.0.11 55555 72.163.4.161 443
```

Review the output to confirm that the packet follows the expected path and is permitted during the simulation. This is useful when you need to determine whether the issue is related to routing, NAT, or policy matching before engaging in live packet analysis.

The transmit keyword in packet-tracer causes the simulated packet to be injected at the ingress interface, enabling it to traverse all firewall processing phases rather than remaining as a simulation.

Example command:

```
FTD1# packet-tracer input Inside tcp 192.168.0.11 55555 72.163.4.161 443 transmit
```

Step 3: Run Captures

Use packet captures to inspect live traffic as it passes through the firewall. Packet captures are different from packet-tracer because they capture actual traffic and therefore must match the real flow that is traversing the

firewall at the time the capture is active. Packet captures are bidirectional, documenting both sides of the connection.

For most traffic troubleshooting scenarios, configure the these captures:

- Ingress Capture — captures traffic arriving on the source-side interface
- Egress Capture — captures traffic leaving on the destination-side interface
- ASP Drop Capture — captures packets that are dropped by the firewall and match the configured criteria

General capture syntax:

```
capture <name> interface <interface name> trace match ip host <source IP> host <destination IP>
```

Ingress capture example:

```
FTD1# capture in interface trace Inside match ip host 192.168.0.11 host 72.163.4.161
```

Egress capture example:

```
FTD1# capture out interface trace Outside match ip host 128.107.0.22 host 72.163.4.161
```

ASP drop capture example:

```
FTD1# cap asp type asp-drop match ip host 192.168.0.11 host 72.163.4.161
```

Packet captures are bidirectional, which means both the forward and return traffic can be captured based on the defined match criteria. The trace feature allows for a visualization of the decisions the Firewall makes on the actual traffic.

Caution: If NAT is being performed, the egress capture must use the translated source IP, not the original source IP, to capture the post-NAT traffic correctly.

Step 4: Run System Support Trace

Use system support trace to view the real-time Snort inspection process for a specific traffic flow. This is especially useful when traffic matches an ACL permit rule but is still blocked by policy inspection. Standard packet captures show that a packet was blocked, but system support trace provides visibility into why the packet was handled that way. System support trace is bidirectional, meaning it receives traffic from both sides. This means the order in which the IPs are added into the tool does not matter.

Run this command and respond to the prompts:

```
> system support trace


Enable firewall-engine-debug too? [n]: y
Please specify an IP protocol: tcp
Please specify a client IP address: 192.168.0.11
Please specify a client port:
Please specify a server IP address: 72.163.4.161
Please specify a server port: 443
```

This tool is particularly helpful when the environment is using application or URL rules, identity-based rules, file policies, or intrusion policies. These features are evaluated after the ACL match, so a packet can be allowed by the access control policy but still blocked by Snort inspection.

Note: The client port can be left blank if the exact source port is unknown.

Verify

Use the output from the route lookup, packet-tracer simulation, packet captures, and system support trace together to confirm the behavior of the traffic flow. The goal is to determine whether the packet is taking the correct path, whether the firewall is permitting or denying it, and whether Snort is blocking it based on deeper inspection.

A complete troubleshooting flow typically confirms:

- The route lookup shows the logical ingress and egress interfaces.
- Packet-tracer confirms the intended forwarding path and policy evaluation.
- Packet captures confirm whether the traffic is arriving and leaving as expected.
- System support trace confirms whether Snort or policy inspection is causing the block.

Troubleshoot

When a traffic issue persists, review the these areas:

- Check whether the route lookup matches the expected source and destination interfaces.
- Verify whether the packet-tracer output shows the traffic being denied at the ACL or NAT stage.
- Review the ingress and egress packet captures to confirm that traffic is reaching the firewall and leaving on the correct interface.
- Use ASP drop captures to confirm whether the firewall is dropping the traffic internally.
- Use system support trace to determine whether Snort is blocking the traffic after the ACL permit decision.

Related Information

- **Analyze Firepower Firewall Captures to Troubleshoot Network Issues**
- **Use Firepower Threat Defense Captures and Packet Tracer**