Collect and Troubleshoot Packet Captures on Cisco ESA

Contents

Introduction

Components Used

Important Risks and Prerequisites

Background Information

Configure Packet Captures on AsyncOS

- Start or Stop a Packet Capture
- Managing Packet Captures
- Packet Capture Constraints
- Additional Support for Packet Captures

Use Custom Packet Capture Filters

Perform Additional Network Investigation

- TCPSERVICES
- NETSTAT
- NETWORK
- ETHERCONFIG
- TRACEROUTE
- PING

Introduction

This document describes how to configure and collect packet captures on the Cisco Email Security Appliance (ESA) for network troubleshooting.

Components Used

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.

Important Risks and Prerequisites

- Packet capture commands can cause the ESA disk space to fill up and can cause performance degradation.
- Cisco recommends that you only use these commands with the assistance of a Cisco TAC Engineer.
- Ensure that you have administrative access to the CLI or GUI of the ESA.

Background Information

Cisco Technical Support can ask you to provide insight into the outbound and inbound network activity of

the ESA. The appliance provides the ability to intercept and display TCP, IP, and other packets that are transmitted or received over the network to which the appliance is attached. Run a packet capture to debug the network setup or to verify the network traffic that reaches or leaves the appliance.

Configure Packet Captures on AsyncOS

This section describes the packet capture process.

Start or Stop a Packet Capture

1. To start a packet capture from the GUI, navigate to the **Help and Support** menu at the top right, choose **Packet Capture**, and then click **Start Capture**.
    1. Alternatively, click **Edit Settings** to specify the IP address(es) and port(s) you want to capture, and then click **Submit**.
    2. Port numbers and IP addresses can be entered using a CSV format (for example: 80, 443). To capture *ANY* port or IP, leave the field(s) blank.
2. To stop the packet capture process, click **Stop Capture**.
    1. A capture that begins in the GUI is preserved between sessions.

1. To start a packet capture from the CLI, enter the **packetcapture > start** command.
    1. Alternatively, use the **setup** command to specify the IP address(es) and port(s) you want to capture.
2. To stop the packet capture process, enter the **packetcapture > stop** command.
    1. The ESA stops the packet capture when the session ends.

Managing Packet Captures

To manage your files, navigate to **Help and Support > Packet Capture** in the GUI. From this page, you can:

- **Monitor Progress:**View real-time statistics for active captures, including current file size and elapsed time.
- **Download Files:**Select a completed capture and click**Download File**to save it to your local machine.
- **Delete Files:**To free up space, select one or more files and click**Delete Selected Files**.

Packet Capture Constraints

- **Single Instance:**Only one packet capture can run at a time.
- **Interface Independence:**The GUI and CLI operate independently regarding packet captures. The GUI only displays and manages captures initiated through the web interface, while the CLI only displays the status of captures started via the command line.

Additional Support for Packet Captures

For more detailed instructions, access the **AsyncOS Online Help**:

1. Navigate to**Help and Support > Online Help**.
2. Search for**Packet Capture**.
3. Select**Running a Packet Capture**.

Use Custom Packet Capture Filters

This section provides information regarding custom capture filters and provides examples.

These are the standard filters used:

- **ip** - Filters for all IP protocol traffic
- **tcp** - Filters for all TCP protocol traffic
- **ip host** - Filters for a specific IP address source or destination

These are examples of the filters in use:

- **ip host 10.1.1.1** - This filter captures any traffic that includes 10.1.1.1 as a source or destination.
- **ip host 10.1.1.1 or ip host 10.1.1.2** - This filter captures traffic that contains either 10.1.1.1 or 10.1.1.2 as a source or destination.

Perform Additional Network Investigation

The methods described below can only be utilized from the CLI.

TCPSERVICES

The **tcpservices** command displays TCP/IP information for current feature and system processes.

```
<#root>

example.com>

tcpservices


System Processes (Note: All processes can not always be present)
  ftpd.main    - The FTP daemon
  ginetd       - The INET daemon
  interface    - The interface controller for inter-process communication
  ipfw         - The IP firewall
  slapd        - The Standalone LDAP daemon
  sntpd        - The SNTP daemon
  sshd         - The SSH daemon
  syslogd      - The system logging daemon
  winbindd     - The Samba Name Service Switch daemon

Feature Processes
  euq_webui    - GUI for ISQ
  gui          - GUI process
  hermes       - MGA mail server
  postgres     - Process for storing and querying quarantine data
  splunkd      - Processes for storing and querying Email Tracking data

COMMAND      USER         TYPE NODE   NAME
postgres     pgsql        IPv4 TCP    127.0.0.1:5432
interface    root         IPv4 TCP    127.0.0.1:53
ftpd.main    root         IPv4 TCP    10.0.202.7:21
gui          root         IPv4 TCP    10.0.202.7:80
gui          root         IPv4 TCP    10.0.202.7:443
ginetd       root         IPv4 TCP    10.0.202.7:22
java         root         IPv6 TCP    [::127.0.0.1]:18081
hermes       root         IPv4 TCP    10.0.202.7:25
hermes       root         IPv4 TCP    10.0.202.7:7025
api_serve    root         IPv4 TCP    10.0.202.7:6080
```

```
api_serve    root         IPv4 TCP    127.0.0.1:60001
api_serve    root         IPv4 TCP    10.0.202.7:6443
nginx        root         IPv4 TCP    *:4431
nginx        nobody       IPv4 TCP    *:4431
nginx        nobody       IPv4 TCP    *:4431
java         root         IPv4 TCP    127.0.0.1:9999
```

NETSTAT

This utility displays network connections for TCP (both incoming and outgoing), routing tables, and a number of network interface and network protocol statistics.

<#root>

example.com>

netstat

```
Choose the information you want to display:
1. List of active sockets.
2. State of network interfaces.
3. Contents of routing tables.
4. Size of the listen queues.
5. Packet traffic information.
```

Example of Option 1 (List of active sockets)

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        (state)
tcp4       0      0 10.0.202.7.10275      10.0.201.4.6025        ESTABLISHED
tcp4       0      0 10.0.202.7.22         10.0.201.4.57759       ESTABLISHED
tcp4       0      0 10.0.202.7.10273      a96-17-177-18.deploy.static.akamaitechnologies.com.80  TIME_
tcp4       0      0 10.0.202.7.10260      10.0.201.5.443     ESTABLISHED
tcp4       0      0 10.0.202.7.10256      10.0.201.5.443     ESTABLISHED
```

Example of Option 2 (State of network interfaces)

```
Show the number of dropped packets? [N]> y

Name    Mtu Network       Address              Ipkts Ierrs Idrop     Ibytes    Opkts Oerrs     Obytes  C
Data 1    - 10.0.202.0    10.0.202.7        110624529     -     - 117062552515 122028093     - 30126949
```

Example of Option 3 (Contents of routing tables)

```
Routing tables

Internet:
Destination        Gateway            Flags      Netif Expire
default            10.0.202.1         UGS         Data 1
10.0.202.0         link#2             U           Data 1
10.0.202.7         link#2             UHS         lo0
localhost.example. link#4             UH          lo0
```

Example of Option 4 (Size of the listen queues)

```
Current listen queue sizes (qlen/incqlen/maxqlen)
Proto Listen         Local Address
tcp4  0/0/50         localhost.exampl.9999
tcp4  0/0/50         10.0.202.7.7025
tcp4  0/0/50         10.0.202.7.25
tcp4  0/0/15         10.0.202.7.6443
tcp4  0/0/15         localhost.exampl.60001
tcp4  0/0/15         10.0.202.7.6080
tcp4  0/0/20         localhost.exampl.18081
tcp4  0/0/20         10.0.202.7.443
tcp4  0/0/20         10.0.202.7.80
tcp4  0/0/10         10.0.202.7.21
tcp4  0/0/10         10.0.202.7.22
tcp4  0/0/10         localhost.exampl.53
tcp4  0/0/208        localhost.exampl.5432
```

Example of Option 5 (Packet traffic information)

```
            input           nic1           output
   packets  errs idrops      bytes    packets  errs      bytes colls drops
        49     0     0       8116         55     0       7496     0     0
```

NETWORK

The network sub-command under diagnostic provides access to additional options.

Use this command to flush all network-related caches, show contents of the ARP cache, show contents of the NDP cache (if applicable), and test remote SMTP connectivity using SMTPPING.

```
<#root>

example.com>

diagnostic



Choose the operation you want to perform:
- RAID - Disk Verify Utility.
- DISK_USAGE - Check Disk Usage.
- NETWORK - Network Utilities.
- REPORTING - Reporting Utilities.
- TRACKING - Tracking Utilities.
- RELOAD - Reset configuration to the initial manufacturer values.
- SERVICES - Service Utilities.
[]>

network



Choose the operation you want to perform:
- FLUSH - Flush all network related caches.
- ARPSHOW - Show system ARP cache.
- NDPSHOW - Show system NDP cache.
```

```
- SMTPPING - Test a remote SMTP server.
- TCPDUMP - Dump ethernet packets.
[]>
```

ETHERCONFIG

The **etherconfig** command allows you to view and configure settings related to duplex and MAC information for interfaces, VLANs, loopback interfaces, MTU sizes, and acceptance or rejection of ARP replies with a multicast address.

<#root>

example.com>

etherconfig

```
Choose the operation you want to perform:
- MEDIA - View and edit ethernet media settings.
- VLAN - View and configure VLANs.
- LOOPBACK - View and configure Loopback.
- MTU - View and configure MTU.
- MULTICAST - Accept or reject ARP replies with a multicast address.
[]>
```

TRACEROUTE

This command displays the network route to a remote host.

Use the **traceroute6** command if you have an IPv6 address configured on at least one interface.

<#root>

example.com>

traceroute google.com

```
Press Ctrl-C to stop.
traceroute to google.com (216.58.194.206), 64 hops max, 40 byte packets
1 68.232.129.2 (68.232.129.2) 0.902 ms
68.232.129.3 (68.232.129.3) 0.786 ms 0.605 ms
2 139.138.24.10 (139.138.24.10) 0.888 ms 0.926 ms 1.092 ms
3 68.232.128.2 (68.232.128.2) 1.116 ms 0.780 ms 0.737 ms
4 139.138.24.42 (139.138.24.42) 0.703 ms
208.90.63.209 (208.90.63.209) 1.413 ms
139.138.24.42 (139.138.24.42) 1.219 ms
5 svl-edge-25.inet.qwest.net (63.150.59.25) 1.436 ms 1.223 ms 1.177 ms
6 snj-edge-04.inet.qwest.net (67.14.34.82) 1.838 ms 2.086 ms 1.740 ms
7 108.170.242.225 (108.170.242.225) 1.986 ms 1.992 ms
108.170.243.1 (108.170.243.1) 2.852 ms
8 108.170.242.225 (108.170.242.225) 2.097 ms
108.170.243.1 (108.170.243.1) 2.967 ms 2.812 ms
9 108.170.237.105 (108.170.237.105) 1.974 ms
```

```
sfo03s01-in-f14.1e100.net (216.58.194.206) 2.042 ms 1.882 ms
```

PING

Ping allows you to test the reachability of a host using either the IP address or hostname and provides statistics related to possible latency and drops in communication.

<#root>

```
example.com>
```

ping google.com

```
Press Ctrl-C to stop.
PING google.com (216.58.194.206): 56 data bytes
64 bytes from 216.58.194.206: icmp_seq=0
```

ttl=56 time=2.095 ms

```
64 bytes from 216.58.194.206: icmp_seq=1 ttl=56 time=1.824 ms
64 bytes from 216.58.194.206: icmp_seq=2 ttl=56 time=2.005 ms
64 bytes from 216.58.194.206: icmp_seq=3 ttl=56 time=1.939 ms
64 bytes from 216.58.194.206: icmp_seq=4 ttl=56 time=1.868 ms
64 bytes from 216.58.194.206: icmp_seq=5 ttl=56 time=1.963 ms

--- google.com ping statistics ---
```

6 packets transmitted, 6 packets received, 0.0% packet loss

round-trip min/avg/max/stddev = 1.824/1.949/2.095/0.088 ms