Troubleshooting

This chapter describes how to troubleshoot the ASA and ASAv and includes the following sections:

Viewing Debugging Messages

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to use **debug** commands during periods

of less network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use. To enable debugging messages, see the **debug** commands in the command reference.

Capturing Packets

Capturing packets may be useful when troubleshooting connectivity problems or monitoring suspicious activity. We recommend that you contact Cisco TAC if you want to use the packet capture feature.

To capture packets, enter the following command:

Command	Purpose
[**cluster exec**] **capture** *capture_name* [**type** {**asp-drop** *drop-code* | **raw-data** | **isakmp** [**ikev1** | **ikev2**] | **tls-proxy** | **lacp** | **webvpn user** *user_name* [**form-only**]}] [**access-list** *acl_name*] [**buffer** *buf-size*] [**ethernet-type** *type*] {**interface** {*if-name* | **asa_dataplane** | **cluster**}} [**packet-length** *bytes*] [**circular-buffer**] [**headers-only**] [**match** *protocol* {**host** *source_ip* | *source_ip mask* | **any**} [*operator port*] {**host** *dest_ip* | *dest_ip mask* | **any**} [*operator port*]] [**real-time** [**dump**] [**detail**] [**trace**]] [**reinject-hide**] [**trace** [**detail**] [**trace-count** *number*]] **Example:** `ciscoasa# capture captest interface inside`	Enables packet capture capabilities for packet sniffing and network fault isolation. For the complete syntax description, see the command reference or the CLI help (**help capture**). Not all options can be specified in one command. See the CLI help for allowed combinations. Use the same *capture_name* on multiple **capture** statements to capture multiple types of traffic. The **type asp-drop** keyword captures packets dropped by the accelerated security path. In a cluster, dropped forwarded data packets from one unit to another are also captured. In multiple context mode, when this option is issued in the system, all context dropped data packets are captured. The **buffer** keyword defines the buffer size used to store the packet. When the byte buffer is full, packet capture stops. When used in a cluster, this is the per-unit size, not the sum of all units. The **circular-buffer** keyword overwrites the buffer, starting from the beginning, when the buffer is full. The **interface** keyword sets the name of the interface on which to use packet capture. You must configure an interface for any packets to be captured. To capture packets on the dataplane, use the **asa_dataplane** keyword. To filter packets captured on the ASA CX backplane, use the **asa_dataplane** option and follow these guidelines. In single mode, the backplane control packets bypass the access list and are captured. In multiple context mode, only control packets are captured in the system context. Data packets are captured in the user context. The **access-list** and **match** options are only available in the user context. To capture the traffic on the cluster control link, use the **cluster** keyword. If you configure **type lacp**, specify the physical interface ID instead of the nameif name. The **match** keyword captures matching the protocol and source and destination IP addresses and optional ports. You can use this keyword up to three times in one command. The *operator* can be as follows: • **lt—**less than • **gt—**greater than • **eq—**equal to The **type raw-data** keywords capture inbound and outbound packets. This setting is the default. The **real-time** keyword displays the captured packets continuously in real-time. To terminate real-time packet capture, enter **Ctrl + c.** To permanently remove the capture, use the **no** form of this command. This option applies only to **raw-data** and **asp-drop** captures. This option is not supported when you use the **cluster exec capture** command. The **reinject-hide** keyword specifies that no reinjected packets will be captured and applies only in a clustering environment. **Note** If ACL optimization is configured, you cannot use the **access-list** command in capture. You can only use the **access-group** command. An error appears if you try to use the **access-list** command in this case.

Capturing Packets in a Clustering Environment

To support cluster-wide troubleshooting, you can enable capture of cluster-specific traffic on the master unit using the **cluster exec capture** command, which is then automatically enabled on all of the slave units in the cluster. The **cluster exec** keywords are the new keywords that you place in front of the **capture** command to enable cluster-wide capture.

The “cluster” interface name is the default name for the cluster control link and is not configurable. You specify “cluster “ as the interface name to capture the traffic on the cluster control link interface. There are two types of packets on the cluster control link: control plane packets and data plane packets, which both include forwarded data traffic and cluster LU messages. The TTL field in the IP address header is encoded to differentiate between these two types of packets. When forwarded data packets are captured, their clustering trailers are included in the capture file for debugging purposes.

In multiple context mode, although the cluster interface belongs to the system context, you can see the interface, so you can configure captures on the cluster link in user contexts. In the system context, both control plane and data plane packets are available. The data plane captures LU packets and forwarded data packets that belong only to the system context. In user contexts, control plane packets are not visible. Only forwarded data packets that belong to a specified user context and LU packets are captured. For security purposes, each context can only see the packets that belong to it.

Guidelines and Limitations

This section includes the guidelines and limitation for this feature.

Most of the limitations are the result of the distributed nature of the ASA architecture and the hardware accelerators that are being used in the ASA.

- You can only capture IP traffic; you cannot capture non-IP packets such as ARPs.
- For cluster control link capture in multiple context mode, only the packet that is associated with the context sent in the cluster control link is captured.
- In multicontext mode, the **copy capture** command is available only in the system space. The syntax is as follows:

  copy /pcap capture:*Context-name*/*in-cap* **tftp**:

  Where *in-cap* is the capture configured in the context *context-name*
- The **cluster exec capture realtime** command is not supported. The following error message appears:

  ```
  Error: Real-time capture can not be run in cluster exec mode.
  ```

- For a shared VLAN, the following guidelines apply:
  - You can only configure one capture for the VLAN; if you configure a capture in multiple contexts on the shared VLAN, then only the last capture that was configured is used.
  - If you remove the last-configured (active) capture, no captures become active, even if you have previously configured a capture in another context; you must remove the capture and add it again to make it active.
  - All traffic that enters the interface to which the capture is attached is captured, including traffic to other contexts on the shared VLAN.
  - Therefore, if you enable a capture in Context A for a VLAN that is also used by Context B, both Context A and Context B ingress traffic are captured.

- For egress traffic, only the traffic of the context with the active capture is captured. The only exception is when you do not enable the ICMP inspection (therefore the ICMP traffic does not have a session in the accelerated path). In this case, both ingress and egress ICMP traffic for all contexts on the shared VLAN is captured.
- Configuring a capture typically involves configuring an ACL that matches the traffic that needs to be captured. After an ACL that matches the traffic pattern is configured, then you need to define a capture and associate this ACL to the capture, along with the interface on which the capture needs to be configured.

After you have performed a cluster-wide capture, to copy the same cluster-wide capture file to a TFTP server, enter the following command on the master unit:

```
ciscoasa (cfg-cluster)# cluster exec copy /pcap capture: cap_name
tftp://location/path/filename.pcap
```

Multiple PCAP files, one from each unit, are copied to the TFTP server. The destination capture file name is automatically attached with the unit name, such as filename_A.pcap, filename_B.pcap, and so on. In this example, A and B are cluster unit names. A different destination name is generated if you add the unit name at the end of the filename.

To enable cluster-wide capture on a specified interface, you can add the **cluster exec** keywords in front of each of the commands shown in the examples. These **capture** commands can only be replicated from the master unit to the slave units. However, you can still configure a capture on the specified interface for the local unit using any of these **capture** commands.

Examples

The following example shows how to create a cluster-wide LACP capture:

```
ciscoasa (config)# cluster exec capture lacp type lacp interface gigabitEthernet0/0
```

The following example shows how to create a capture for control path packets in the clustering link:

```
ciscoasa (config)# capture cp interface cluster match udp any eq 49495 any
ciscoasa (config)# capture cp interface cluster match udp any any eq 49495
```

The following example shows how to create a capture for data path packets in the clustering link:

```
ciscoasa (config)# access-list cc1 extended permit udp any any eq 4193
ciscoasa (config)# access-list cc1 extended permit udp any eq 4193 any
ciscoasa (config)# capture dp interface cluster access-list cc1
```

The following example shows how to capture data path traffic through the cluster:

```
ciscoasa (config)# capture abc interface inside match tcp host 1.1.1.1 host 2.2.2.2 eq www
ciscoasa (config)# capture abc interface inside match udp host 1.1.1.1 any
ciscoasa (config)# capture abc interface inside access-list xxx
```

The following example shows how to capture logical update messages for flows that match the real source to the real destination, and capture packets forwarded over CCL that match the real source to the real destination:

```
ciscoasa (config)# access-list dp permit ip real_src real_dst
```

The following example shows how to capture a certain type of data plane message, such as icmp echo request/response, that is forwarded from one ASA to another ASA using the **match** keyword or the ACL for the message type:

```
ciscoasa (config)# capture capture_name interface cluster access-list match icmp any any
```

The following example shows how to create a capture by using ACL 103 on a cluster control link:

```
ciscoasa (config)# access-list 103 permit ip A B
ciscoasa (config)# capture example1 interface cluster access-list 103
```

In the previous example, if A and B are IP addresses for the CCL interface, only the packets that are sent between these two units are captured.

If A and B are IP addresses for through-device traffic, then the following is true:

- Forwarded packets are captured as usual, provided the source and destination IP addresses are matched with the ACL.
- The data path logic update message is captured provided it is for the flow between A and B or for an ACL (for example, access-list 103). The capture matches the five-tuple of the embedded flow.
- Although the source and destination addresses in the UDP packet are CCL addresses, if this packet is to update a flow that is associated with addresses A and B, then it is also captured. That is, as long as addresses A and B that are embedded in the packet are matched, it is also captured.

For more information about clustering, see Chapter 9, “ASA Cluster.”

Viewing the Crash Dump

If the ASA or ASAv crashes, you can view the crash dump information. We recommend that you contact Cisco TAC if you want to interpret the crash dump. See the **show crashdump** command in the command reference.

Viewing the Coredump

A coredump is a snapshot of the running program when the program has terminated abnormally or crashed. Coredumps are used to diagnose or debug errors and save a crash for future off-site analysis. Cisco TAC may request that you enable the coredump feature to troubleshoot application or system crashes on the ASA or ASAv. See the **coredump** command in the command reference.

vCPU Usage in the ASAv

The ASAv vCPU usage shows the amount of vCPUs used for the data path, control point, and external processes.

The vSphere reported vCPU usage includes the ASAv usage as described plus:

- ASAv idle time
- %SYS overhead used for the ASAv VM
- Overhead of moving packets between vSwitches, vNICs, and pNICs. This overhead can be quite significant.

CPU Usage Example

The following is an example in which the reported vCPU usage is substantially different:

- ASAv reports: 40%

- DP: 35%
- External Processes: 5%
- vSphere reports: 95%
- ASA (as ASAv reports): 40%
- ASA idle polling: 10%
- Overhead: 45%

The overhead is used to perform hypervisor functions and to move packets between NICs and vNICs using the vSwitch.

Usage can exceed 100% because the ESXi server can use additional compute resources for overhead on behalf of the ASAv.

VMware CPU Usage Reporting

In vSphere, click the **VM Performance** tab, then click **Advanced** to display the Chart Options drop-down list, which shows vCPU usage for each state (%USER, %IDLE, %SYS, and so on) of the VM. This information is useful for understanding VMware's perspective on where CPU resources are being used.

On the ESXi server shell (you access the shell by using SSH to connect to the host), esxtop is available. Esxtop has a similar look and feel to the Linux **top** command and provides VM state information for vSphere performance, including the following:

- Details on vCPU, memory, and network usage
- vCPU usage for each state of each VM.
- Memory (type M while running) and network (type N while running), as well as statistics and the number of RX drops

ASAv and vCenter Graphs

There are differences in the CPU % numbers between the ASAv and vCenter:

- The vCenter graph numbers are always higher than the ASAv numbers.
- vCenter calls it %CPU usage; the ASAv calls it %CPU utilization.

The terms "%CPU utilization" and "%CPU usage" mean different things:

- CPU utilization provides statistics for physical CPUs.
- CPU usage provides statistics for logical CPUs, which is based on CPU hyperthreading. But because only one vCPU is used, hyperthreading is not turned on.

vCenter calculates the CPU % usage as follows:

Amount of actively used virtual CPUs, specified as a percentage of the total available CPUs

This calculation is the host view of the CPU usage, not the guest operating system view, and is the average CPU utilization over all available virtual CPUs in the virtual machine.

For example, if a virtual machine with one virtual CPU is running on a host that has four physical CPUs and the CPU usage is 100%, the virtual machine is using one physical CPU completely. The virtual CPU usage calculation is as follows:

Usage in MHz / number of virtual CPUs x core frequency

When you compare the usage in MHz, both the vCenter and ASAv numbers match. According to the vCenter graph, MHz % CPU usage is calculated as:

60/(2499 x 1 vCPU) = 2.4