Configuring Cores

Core files can be saved either manually by a user or automatically at the time of the fault. If a core file is created, preserve it by copying it to nonvolatile file space (such as to a host) and report it to Cisco for diagnosis.

Cores may be copied multiple times. Both IPv4, IPv6, and many protocols are supported for copying cores to file space on remote hosts. This includes passwordless SSH which is convenient for automatic copying in secure environments. For more information about configuring passwordless access to remote hosts, see the 'Passwordless File Copy and SSH' section in the 'Configuring SSH Services and Telnet' chapter of Cisco Nexus 7000 Series NX-OS Security Configuration Guide.

There is no upper limit on the total number of core files in the active supervisor module.

Tip Before copying a core, ensure that you create the destination directory with write permission for the user.

Guideline and Limitations

- Kernel Core Collection and HW Watch for SDB4 are supported in Cisco Nexus 7700 switches only.
- HW Watch for SDB4 descriptor configuration is not stored in running-config and not retained during switchover, reload or ISSU.

Configuring Kernel Core Collection

To configure kernel core collection, follow these steps:

Procedure

Step 1 switch# **configure terminal**

Enters configuration mode.

Step 2 switch(config)# **system kernel core**

Enables collection of kernel core if there is a kernel crash.

Step 3 switch(config)# **no system kernel core**

(Optional) Disables collection of kernel cores.

Topic 2.1

Copying Cores Manually

The supported on-switch destination is slot0. Supported protocols to transfer cores to remote destinations are TFTP, SFTP, and SCP.

To configure saving cores manually, follow these steps:

Procedure

switch# **copy** *core://module/process-id[/instance] destination://[[user@]host/][directory]*

Copies a core of process to the specified location.

Copying Cores Automatically

Supported on-switch destinations are bootflash, slot0, and usb1. Supported protocols to transfer cores to remote destinations are HTTP, HTTPS, TFTP, FTP, SFTP, and SCP.

To configure saving cores automatically, follow these steps:

Procedure

Step 1 switch# **configure**

Enters configuration mode.

Step 2 switch(config)# **system cores** *destination://[[user@]host/][directory]*

Save core files to the specified destination as soon as they are created.

Step 3 switch(config)# **no system cores**

(Optional) Disables saving core files automatically.

Deleting Cores

Core files are not automatically deleted after copying. After a core is copied, delete it from the switch core repository to reclaim the space and report it to Cisco support for analysis.

Use the **clear core_file** command to delete a single core from the switch core repository.

switch# **clear core_file module** *module* **pid** *pid*

Use the **clear cores** command to clear all cores in the switch core repository.

switch# **clear cores**

Configuring Hardware Watch for SDB4 Descriptor

To configure hardware watch for SDB4 descriptor, follow these steps:

Procedure

Step 1 switch# **system kernel sdb4_ext3_desc_watch**

Enables hardware watch for SDB4 descriptor when there is a kernel crash.

Step 2 switch# **no system kernel sdb4_ext3_desc_watch**

(Optional) Disables harware watch for SDB4 descriptor.

Step 3 switch# **show system kernel sdb4_ext3_desc_watch**

Displays the status of hardware watch for SDB4 descriptor.

Example: Configuring Cores

The following example copies a core of process with PID 1524 generated on slot 5 to the *cores* directory on a host with HTTPS as user *n7kadmin*:

```
switch# copy core://5/1524 https://n7kadmin@192.168.1.2/cores
```

The following example automatically copies any core files immediately after they are created to the /tftpboot/cores directory on a host with SCP as user mdsadmin. Configure passwordless SSH first for this to work.

```
switch# configure
switch(config)# system cores scp://n7kadmin@192.168.1.2/tftpboot/cores
```

The following example deletes the core generated from module 1 for the process with PID 1234.

```
switch# clear core_file module 1 pid 1234
```