Troubleshoot Unexpected Reloads in IOS Platforms with TAC

Contents

Introduction

Prerequisites

- Requirements

Show Tech-Support Files

- Log a Terminal Session
- Create a File in Storage

Crashinfo File

Core Files

Tracelogs

System Reports

Kernel Cores

How to Extract Files

- TFTP
- FTP
- SCP
- USB

Troubleshoot

- Confirm Open Ports
- USB Format
- Transfer Interruptions
- Intermediate TFTP Server

Introduction

This document describes files required to determine the cause of an unexpected reload in Cisco IOS®/ XE and upload them to a TAC case.

Prerequisites

Requirements

- This document applies to Cisco routers and switches that run Cisco IOS®/Cisco IOS® XE software.
- In order to collect the files described in this document, the device has to be up and stable.
- In order to extract the files via transfer protocol, a server (with file transfer application/service installed) with L3 reachability is required.
- Console or remote connection via SSH/Telnet to the device is needed.
- SD-WAN deployments are not discussed.

Note: In an unexpected reload event, it is possible that some files are not generated based on the nature of the reload and the platform.

Show Tech-Support Files

The **show tech-support** command output includes general information about the device current status (memory and CPU utilization, logs, configuration, etc), and information about the created files related to when the unexpected reload event took place.

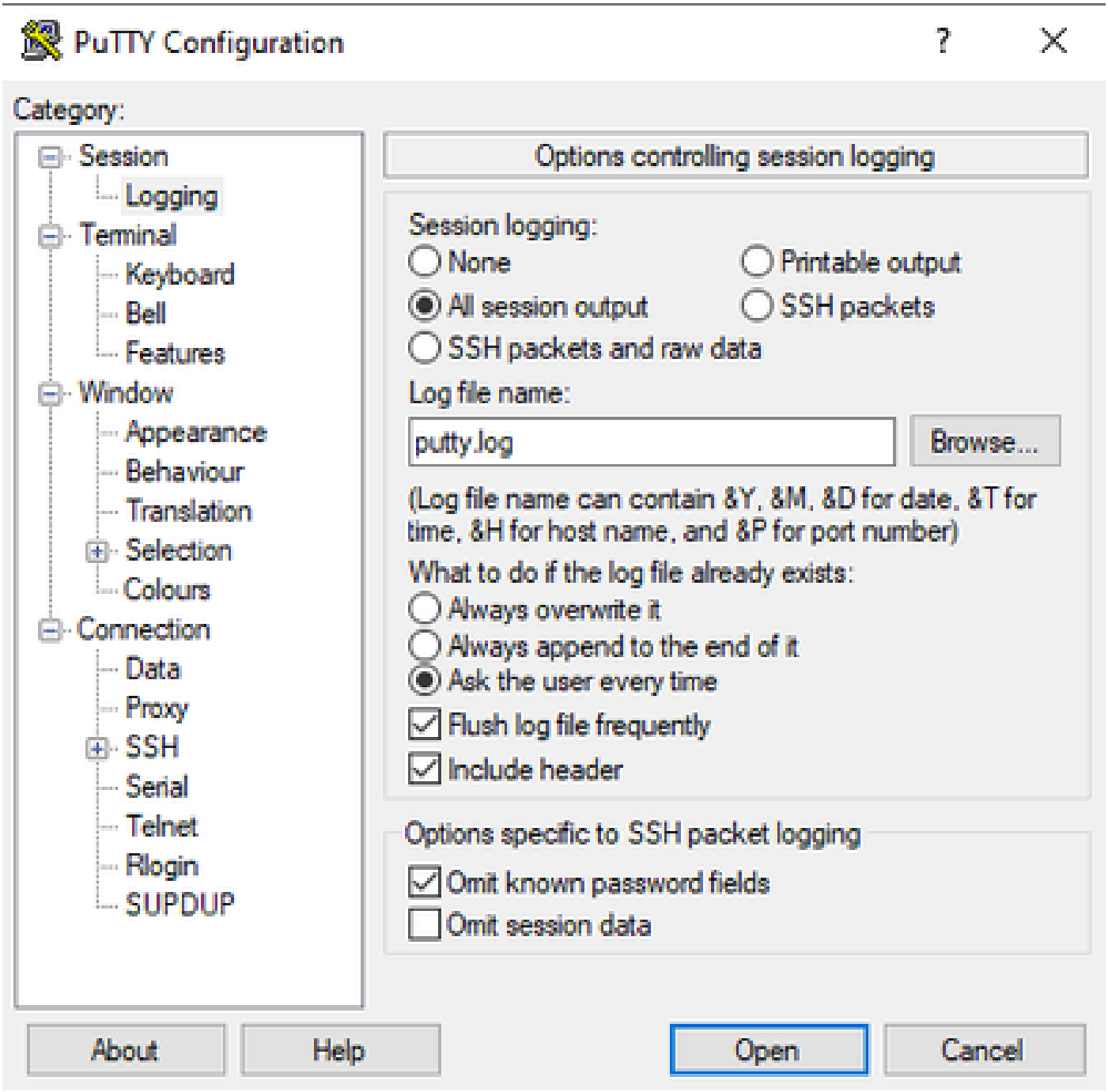
In the case of an unexpected reboot situation, the key points to review are:

- The current Cisco IOS/Cisco IOS XE version installed on the device.
- System configuration with ports, cards and modules details.
- Presence of additional files to provide a root cause analysis in the file systems.

The show tech-support output can be captured in two different ways: log a terminal session or create a file in storage and transfer it off the device.

Log a Terminal Session

In Putty, navigate to **Session** > **Logging** and select inside the **Session logging** tab, select the **All session output** option, as shown in this image.



The file is stored in the Putty folder by default with putty.log name. The folder and name of the file can be changed with the **Browse** button.

Once the configuration is completed, the **Putty** session has to be connected to the device via **Console**, **Telnet** or **SSH**.

In the device session, it is advised to set the **terminal length 0** command in privilege mode and then use the **show tech-support** command.

```
# terminal length 0
# show tech-support
```

Note: The command can take a couple of seconds to execute. Do not interrupt the execution.

Create a File in Storage

A **show tech-support** file can be created on the device and stored in one of the file system storage (internal or external). The command syntax remains the same in all the devices but the file system used can be changed. The file can also be created on an external server directly, this section shows the syntax for a local file system.

In order to create the file inside the flash, it is required to use the command **show tech-support | redirect flash:Showtech.txt** in privilege mode:

```
# show tech-support | redirect flash:Showtech.txt
```

The terminal can not be used for a couple of seconds while the text file is generated. Once it is completed, you can verify if the file creation is correct with the **show [file system]:** command; since the file is a plain text file, the content can be displayed on the device with **more** command.

```
# show flash:
# more flash:Showtech.txt
```

Once the file is created, it can be extracted to an external storage with a transfer protocol of choice (FTP/TFTP/SCP) and shared for analysis.

Crashinfo File

The crashinfo file is a text file, it includes debug details that would help to identify the reason for the crash. The content can vary from platform to platform. In general, it has the **logging buffer** prior to the crash and the functions that were run by the processor, prior to the crash in an encoded mode. In Cisco IOS platforms, this is the most common file that can be found in the file systems after the crash. In Cisco IOS XE platforms, this file is generated when the crash happens in the IOSd process only; if any other process fails, then the device does not create a crashinfo file.

Crashinfo files can be found under flash, bootflash, harddisk or crashinfo storage in base on the platform. In the case of redundant control plane platforms, the crashfiles can be found in the active and/or standby supervisor.

The content of this file is limited, as it only takes a snap of the DRAM memory prior to the unexpected reboot and the memory region of the processes. Additional files/outputs can be required to identify the root cause of the reboot in some cases.

Core Files

On Cisco IOS XE platforms, when a process or a service terminates its execution due to a runtime error (and causes an unexpected reboot), a core file is created. This file contains context information about the reload event.

On Cisco IOS XE platforms, it is generated by default when the unexpected reboot is software-driven. The core files can be created under any Linux process (IOSd processes included).

Core files are compressed files that contain the information of all the memory in execution used by the specific process that triggered the crash. This file requires special tools to decode, therefore, to maintain its consistency, it is required to extract the file without any change. Decompress the file, or extract the information as text (such as with **more** command), does not permit the ability to decode the content by the support team.

Core files usually are stored in the **core** folder, inside the bootflash or harddisk.

Next is an example that shows how the corefile appears inside the core folder in the bootflash file system:

```
------------------ show bootflash: all ------------------

 9   10628763 Jul 14 2021 09:58:49 +00:00 /bootflash/core/Router_216_Router_RP_0_ucode_pkt_PPE0_3129_16
10   10626597 Jul 23 2021 13:35:26 +00:00 /bootflash/core/Router_216_Router_RP_0_ucode_pkt_PPE0_2671_16
```

Note: In order for TAC to successfully analyze the corefile, it is required to extract the files without any modification or change.

In order to verify the way to extract this file from the device, navigate to the Extract Files section.

Tracelogs

The tracelogs are internal logs of each process within Cisco IOS XE. The tracelogs directory is created by default and its content is overwritten periodically. This folder can be found in the bootflash or the harddisk.

The folder can be safely removed, although it is not recommended as it can provide additional information in the case of an unexpected reload event.

Depending on the state of the troubleshooting it would be necessary to overwrite the oldest tracelogs files with the latest one or collect the oldest ones. If latest tracelogs are needed it would be used the command:

```
# request platform software trace rotate all
```

In order to extract the content of the folder, the easiest approach is to create a compressed file that includes all the tracelogs files. In base on the platform, you can use these commands:

For Cisco IOS XE routers:

```
# request platform software trace slot rp active archive target bootflash:TAC_tracelogs
```

For Cisco IOS XE switches and wireless controllers:

```
# request platform software trace archive target bootflash:TAC_tracelogs
```

Tracelogs are encoded files that require additional tools to decode, thus it is required to extract the compressed file as it is created.

To check the way to extract this file from the device, navigate to the Extract Files section.

System Reports

A system report is a compressed file that collects most of the information available in the software execution when an unexpected reload occurs. The system report contains tracelogs, crashinfo, and core files. This file is created in the case of an unexpected reload on Cisco IOS XE switches and wireless controllers.

The file can be found in the main directory of the bootlfash or harddisk.

It always contains the tracelogs generated just before the reboot. In the case of an unexpected reload it has crashfiles and core files of the event.

This file is a compressed file, the folder can be decompressed but it requires additional tools to decode the information.

In order to check the way to extract this file from the device, navigate to the Extract Files section.

Kernel Cores

The kernel cores are created by the Linux Kernel and not by Cisco IOS XE processes. When a device reloads because of a kernel failure, usually a complete kernel core (compressed file) and a summary of the kernel core (plain text) files are created.

The processes that led to the unexpected reboot can be reviewed but it is always recommended to provide the file to Cisco TAC in order to provide a complete analysis of the reload reason.

The kernel core files can be found in the main directory of the bootflash or harddisk.

How to Extract Files

This section describes the basic configuration required in order to transfer the required files from the Cisco IOS/Cisco IOS XE platform to an external storage client.

Reachability from the device to the server is expected to be available. If necessary, confirm there is no firewall or configuration that blocks the traffic from the device to the server.

No specific server application is recommended in this section.

TFTP

In order to transfer a file over TFTP, it is required to set reachability to the TFTP server application. No additional configuration is required.

By default, some devices have the **ip tftp source interface** configuration active via the management interface. If the server is not reachable through the management interface, run the command in order to remove this configuration:

```
(config)# no ip tftp source interface
```

Once the configuration to reach the server is done, in order to transfer the file you can run these commands:

```
#copy <storage>:<file> tftp:
Address or name of remote host []? X.X.X.X
Destination filename [<file>]?
```

FTP

In order to transfer a file over FTP, it is required to set reachability to the FTP server application. It is necessary to configure FTP username and password from the device and the FTP server application. In order to set the credentials on the device, run these commands:

```
(config)#ip ftp username username
(config)#ip ftp password password
```

Optionally, you can configure an FTP source interface on the device with these commands:

```
(config)# ip ftp source interface interface
```

Once the configuration to reach the server is complete, in order to transfer the file you can run these command:

```
#copy <storage>:<file> ftp:
Address or name of remote host []? X.X.X.X
Destination filename [<file>]?
```

SCP

In order to transfer a file over SCP, it is required to set reachability to the SCP server application. It is necessary to configure local username and password on the device (credentials are required to start the transfer) and the SCP server application. It is also needed to have SSH configured on the device. In order to confirm the SSH service is configured, run the command:

```
#show running-config | section ssh
ip ssh version 2
ip ssh server algorithm encryption 3des-cbc aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption 3des-cbc aes128-ctr aes192-ctr aes256-ctr
```

```
transport input ssh
transport input ssh
```

In order to set the credentials on the device, run the command:

```
(config)#username USER password PASSWORD
```

Note: In case TACACS or another service is used for SSH user authentication, those credentials can be used if the SCP server also has the user information.

Once the configuration is done, in order to transfer the file you can run these commands:

```
#copy <storage>:<file> scp:
Address or name of remote host []? X.X.X.X
Destination filename [<file>]?
```

USB

The transfer of files through the USB flash does not require reachability to any external server in the network, but it requires physical access to the device.

All the physical devices with Cisco IOS/Cisco IOS XE have USB ports that can be used as external storage.

In order to confirm the USB flash drive is recognized, run the **show file systems** command:

```
#show file systems
File Systems:

    Size(b)     Free(b)     Type     Flags     Prefixes
        -     -         opaque     rw     system:
        -     -         opaque     rw     tmpsys:
*     11575476224     10111098880     disk     rw     bootflash: flash:
    2006351872     1896345600     disk     ro     webui:
        -     -         opaque     rw     null:
        -     -         opaque     ro     tar:
        -     -         network rw     tftp:
    33554432     33527716     nvram     rw     nvram:
        -     -         opaque     wo     syslog:
        -     -         network rw     rcp:
        -     -         network rw     pram:
        -     -         network rw     http:
        -     -         network rw     ftp:
        -     -         network rw     scp:
        -     -         network rw     sftp
        -     -         network rw     https:
        -     -         network ro     cns:
    2006351872    1896345600    disk    rw    usbflash0:
```

Note: Cisco IOS/Cisco IOS XE devices support official Cisco USB flash drives. For any third-party USB flash, support is limited.

Once the USB flash is recognized by the device in the proper slot (usbflash0 or usbflash1) and there is enough free space available, use these commands to transfer the file:

```
#copy <storage>:<file> usbflashX:
Destination filename [<file>]?
```

Troubleshoot

This section describes some of the common errors and workarounds that can be found and used while to transfer files (from an Cisco IOS or Cisco IOS XE device) to an external method.

Confirm Open Ports

If the device shows a connection refused error when the reachability to the server has been confirmed, it can be useful to verify the ports on the device side are available (no ACL entry that blocks the traffic) and that the ports on the server-side are also available. For the last part, the telnet command with the port required can be used.

In base on the protocol used, run these commands:

<#root>

TFTP

```
#telnet X.X.X.X 69
```

FTP

```
#telnet X.X.X.X 21
```

SCP

```
#telnet X.X.X.X 22
```

Note: Previous ports are the default ports for each protocol, it is possible for these ports to be changed.

If the command does not provide a successful open port, it is helpful to confirm any misconfiguration from the server-side or any firewall in the path) that can drop the traffic.

USB Format

Third-party USB can not be recognized for most of the Cisco IOS and Cisco IOS XE devices.

A USB bigger than 4GB can not be recognized by Cisco IOS routers and switches. USB with a size bigger than 4GB can be recognized by Cisco IOS XE platforms.

In the case of a third-party USB, it can be tested with FAT32 or FAT16 formatting. Any other format cannot be recognized even for a compatible USB memory drive.

Transfer Interruptions

It is possible that the file transfer can be interrupted and required to start the transfer again for servers with many hops.

In this scenario, it can be useful to use this configuration on the vty lines:

```
(config)#line vty 0 4
(config-line)#exec-timeout 0 0
```

The previous configuration ensures the transfer session is not dropped, even if the control packet is dropped in the path or the packet takes too long to be acknowledged.

After the transfer is completed, it is recommended to remove this configuration from the vty lines.

It is always recommended to place the file server as close as possible to the device.

Intermediate TFTP Server

The Cisco devices can be used as a temporal TFTP server for transfers that can not be done directly to a local file server.

On the device (with the file that requires extraction), you can run the command:

```
(config)#tftp-server <storage>:<file>
```

From the device that is configured as a client, you can run the commands that appear in the TFTP section.