Troubleshooting

This chapter contains the following sections:

Troubleshooting Overview

This section describes the troubleshooting scenarios.

Before troubleshooting a software problem, you must connect a PC to the router via the console port. With a connected PC, you can view status messages from the router and enter commands to troubleshoot a problem.

You can also remotely access the interface by using Telnet. The Telnet option assumes that the interface is up and running.

Understanding Diagnostic Mode

The router boots up or accesses diagnostic mode in the following scenarios:

- The IOS process or processes fail, in some scenarios. In other scenarios, the system resets when the IOS process or processes fail.

- A user-configured access policy was configured using the **transport-map** command that directs the user into the diagnostic mode.

- A send break signal (**Ctrl-C** or **Ctrl-Shift-6**) was entered while accessing the router, and the router was configured to enter diagnostic mode when a break signal was sent.

In the diagnostic mode, a subset of the commands that are available in user EXEC mode are made available to the users. Among other things, these commands can be used to:

- Inspect various states on the router, including the IOS state.
- Replace or roll back the configuration.
- Provide methods of restarting the IOS or other processes.
- Reboot hardware, such as the entire router, a module, or possibly other hardware components.
- Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

The diagnostic mode provides a more comprehensive user interface for troubleshooting than previous routers, which relied on limited access methods during failures, such as ROMMON, to diagnose and troubleshoot Cisco IOS problems. The diagnostic mode commands can work when the Cisco IOS process is not working properly. These commands are also available in privileged EXEC mode on the router when the router is working normally.

Before Contacting Cisco or Your Reseller

If you cannot locate the source of a problem, contact your local reseller for advice. Before you call, you should have the following information ready:

- Chassis type and serial number
- Maintenance agreement or warranty information
- Type of software and version number
- Date you received the hardware
- Brief description of the problem
- Brief description of the steps you have taken to isolate the problem

show interfaces Troubleshooting Command

Use the **show interfaces** command to display the status of all physical ports and logical interfaces on the router.

The IR1101 supports the following interfaces:

- GigabitEthernet 0/0/0
- Cellular 0/1/0
- FastEthernet 0/0/1 to 0/0/4
- Async 0/2/0
- Cellular 0/x/x

- LORAWAN0/x/0

Change the Configuration Register

To change a configuration register, follow these steps:

Procedure

Step 1 Connect a PC to the CONSOLE port on the router.

Step 2 At the privileged EXEC prompt (*router_name #*), enter the **show version** command to display the existing configuration register value (shown in bold at the bottom of this output example):

Example:

```
Router# show version
Cisco IOS XE Software, Version 16.10.01
Cisco IOS Software [Gibraltar], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9-M), Version 16.10.1,
RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Fri 09-Nov-18 18:08 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2018 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: IOS-XE ROMMON

Router uptime is 14 hours, 36 minutes
Uptime for this control processor is 14 hours, 37 minutes
System returned to ROM by reload
System restarted at 08:47:04 GMT Mon Nov 12 2018
System image file is "bootflash:ir1101-universalk9.16.10.01.SPA.bin"
Last reload reason: Reload Command



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
```

```
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.


Technology Package License Information:

-----------------------------------------------------------------------------
Technology-package                                    Technology-package
Current                    Type                          Next reboot
-----------------------------------------------------------------------------
network-essentials     Smart License                  network-essentials


Smart Licensing Status: UNREGISTERED/EVAL MODE

cisco IR1101-K9 (ARM64) processor (revision 1.2 GHz) with 711861K/6147K bytes of memory.
Processor board ID FCW222700MY
3 Virtual Ethernet interfaces
4 FastEthernet interfaces
1 Gigabit Ethernet interface
1 Serial interface
1 terminal line
2 Cellular interfaces
32768K bytes of non-volatile configuration memory.
4038072K bytes of physical memory.
3110864K bytes of Bootflash at bootflash:.
0K bytes of WebUI ODM Files at webui:.

Configuration register is 0x1821


Router#
```

Step 3 Record the setting of the configuration register.

Step 4 To enable the break setting (indicated by the value of bit 8 in the configuration register), enter the **config-register 0x01** command from privileged EXEC mode.

- Break enabled—Bit 8 is set to 0.
- Break disabled (default setting)—Bit 8 is set to 1.

Configuring the Configuration Register for Autoboot

Note Altering the configuration register is only for advanced troubleshooting and should only be done with guidance from Cisco support.

The configuration register can be used to change router behavior. This includes controlling how the router boots. Set the configuration register to 0x0 to boot into ROM, by using one of the following commands:

- In Cisco IOS configuration mode, use the **config-reg** 0x0 command.
- From the ROMMON prompt, use the **confreg** 0x0 command.

Note: Setting the configuration register to 0x2102 will set the router to autoboot the Cisco IOS XE software.

Reset the Router

To reset the router, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	If break is disabled, turn the router off (O), wait 5 seconds, and turn it on (|) again. Within 60 seconds, press the **Break** key. The terminal displays the ROM monitor prompt.	**Note** Some terminal keyboards have a key labeled *Break* . If your keyboard does not have a Break key, see the documentation that came with the terminal for instructions on how to send a break.
Step 2	Press break. The terminal displays the following prompt: **Example:** `rommon 2>`	
Step 3	Enter **confreg 0x142** to reset the configuration register: **Example:** `rommon 2> confreg 0x142`	

Step 4	Initialize the router by entering the **reset** command:
	Example:
	`rommon 2> reset`
	Example:
	`--- System Configuration Dialog ---`
		The router cycles its power, and the configuration register is set to 0x142. The router uses the boot ROM system image, indicated by the system configuration dialog:

Step 5	Enter **no** in response to the prompts until the following message is displayed: **Example:** `Press RETURN to get started!`	
Step 6	Press **Return**. The following prompt appears: **Example:** `Router>`	
Step 7	Enter the enable command to enter enable mode. Configuration changes can be made only in enable mode:	The prompt changes to the privileged EXEC prompt:

	Command or Action	Purpose

	Example:
	`Router> enable`
	Example:
	`Router#`	

Step 8	Enter the **show startup-config** command to display an enable password in the configuration file: **Example:** `Router# show startup-config`	

What to do next

If you are recovering an enable password, do not perform the steps in the Reset the Password and Save Your Changes section. Instead, complete the password recovery process by performing the steps in the Reset the Configuration Register Value section.

If you are recovering an enable secret password, it is not displayed in the **show startup-config** command output. Complete the password recovery process by performing the steps in the Reset the Password and Save Your Changes section.

Recovering a Lost Password

To recover a lost enable or lost enable-secret password, refer to the following sections:

1. Change the Configuration Register
2. Reset the Router
3. Reset the Password and Save your Changes (for lost enable secret passwords only)
4. Reset the Configuration Register Value.
5. If you have performed a **write erase**, or used the reset button, you will need to add the license.

```
IR1101#config term
IR1101#license smart reservation
```

Note Recovering a lost password is only possible when you are connected to the router through the console port. These procedures cannot be performed through a Telnet session.

Tip See the “Hot Tips” section on Cisco.com for additional information on replacing enable secret passwords.

Reset the Password and Save Your Changes

To reset your password and save the changes, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	Enter the **configure terminal** command to enter global configuration mode: **Example:** `Router# configure terminal`	

Step 2	Enter the **enable secret** command to reset the enable secret password in the router:
	Example:
	`Router(config)# enable secret`
	`password`	

Step 3	Enter **exit** to exit global configuration mode: **Example:** `Router(config)# exit`	
Step 4	Save your configuration changes: **Example:** `Router# copy running-config startup-config`	

Password Recovery Disable

The No Service Password-Recovery is a Cisco IOS Platform independent feature/CLI which is available in Cisco IOS-XE devices. When the No Service Password-Recovery security feature is enabled, it prevents anyone with console access from using a break sequence (Control+C) during bootup to enter into rommon.

Note Ensure a valid Cisco IOS image is present in flash before enabling this feature. Failure to do so will result in the router going into a into boot loop. Hard power reset button is disabled if system has **no service password recovery**.

The following events will cause the router to go into rommon mode as standard IOS-XE behavior:

- config-reg setting is manual boot

- User opts to reset to factory default option

For more information and configuration steps, refer to the following:https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_cfg/configuration/15-sy/sec-usr-cfg-15-sy-book/sec-no-svc-pw-recvry.html

Config register change issue with service password recovery update

When service password recovery is disabled, then the config register cannot be changed and will be stuck at 0x01. This issue was found on the IR1101 Router. For additional information see the tech note Understand Configuration Register Usage on all Routers.

Reset the Configuration Register Value

To reset the configuration register value after you have recovered or reconfigured a password, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	Enter the **configure terminal** command to enter global configuration mode: **Example:** `Router# configure terminal`	
Step 2	Enter the **configure register** command and the original configuration register value that you recorded. **Example:** `Router(config)# config-reg value`	
Step 3	Enter **exit** to exit configuration mode: **Example:** `Router(config)# exit`	**Note** To return to the configuration being used before you recovered the lost enable password, do not save the configuration changes before rebooting the router.
Step 4	Reboot the router, and enter the recovered password.	

Configuring a Console Port Transport Map

This task describes how to configure a transport map for a console port interface on the router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**transport-map type console** *transport-map-name* **Example:** `Router(config)# transport-map type console consolehandler`	Creates and names a transport map for handling console connections, and enters transport map configuration mode.
Step 4	**connection wait** [**allow** [**interruptible**] | **none** [**disconnect**]] **Example:** `Router(config-tmap)# connection wait none`	Specifies how a console connection will be handled using this transport map. • **allow interruptible**—The console connection waits for a Cisco IOS VTY line to become available, and also allows users to enter diagnostic mode by interrupting a console connection that is waiting for a Cisco IOS VTY line to become available. This is the default setting. **Note** Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**. • **none**—The console connection immediately enters diagnostic mode.

Step 5	(Optional) **banner** [**diagnostic** | **wait**] *banner-message*
	Example:
	`Router(config-tmap)# banner diagnostic X`
	`Enter TEXT message. End with the character 'X'.`
	`--Welcome to Diagnostic Mode--`
	`X`
	`Router(config-tmap)#`
		(Optional) Creates a banner message that will be seen by users entering diagnostic mode or waiting for the Cisco IOS VTY line because of the console transport map configuration.
		• **diagnostic**—Creates a banner message seen by users directed to diagnostic mode because of the console transport map configuration.
		Note Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**.
		• **wait**—Creates a banner message seen by users waiting for Cisco IOS VTY to become available.
		• *banner-message*—Banner message, which begins and ends with the same delimiting character.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-tmap)# exit`	Exits transport map configuration mode to re-enter global configuration mode.
Step 7	**transport type console** *console-line-number* **input** *transport-map-name* **Example:** `Router(config)# transport type console 0 input consolehandler`	Applies the settings defined in the transport map to the console interface. The *transport-map-name* for this command must match the *transport-map-name* defined in the **transport-map type console** command.

Examples

The following example shows how to create a transport map to set console port access policies and attach to console port 0:

```
Router(config)# transport-map type console consolehandler
Router(config-tmap)# connection wait allow interruptible
Router(config-tmap)# banner diagnostic X
Enter TEXT message. End with the character 'X'.
--Welcome to diagnostic mode--
X
Router(config-tmap)# banner wait X
Enter TEXT message. End with the character 'X'.
Waiting for IOS vty line
X
Router(config-tmap)# exit
Router(config)# transport type console 0 input consolehandler
```

Viewing Console Port, SSH, and Telnet Handling Configurations

Use the following commands to view console port, SSH, and Telnet handling configurations:

- **show transport-map**
- **show platform software configuration access policy**

Use the **show transport-map** command to view transport map configurations.

show transport-map [**all** | **name** *transport-map-name* | **type** [**console**]]

This command can be used either in user EXEC mode or privileged EXEC mode.

Example

The following example shows transport maps that are configured on the router: console port (`consolehandler`):

```
Router# show transport-map all
Transport Map:
Name: consolehandler Type: Console Transport
```

```
Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type console
Transport Map:
Name: consolehandler


REVIEW DRAFT - CISCO CONFIDENTIAL

Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type persistent ssh
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode
```

Use the **show platform software configuration access policy** command to view the current configurations for handling the incoming console port, SSH, and Telnet connections. The output of this command provides the current wait policy for each type of connection (Telnet, SSH, and console), as well as information on the currently configured banners.

Unlike the **show transport-map** command, the **show platform software configuration access policy** command is available in diagnostic mode so that it can be entered in scenarios where you need transport map configuration information, but cannot access the Cisco IOS CLI.

Example

The following example shows the **show platform software configuration access policy** command.

```
Router# show platform software configuration access policy
The current access-policies

Method : telnet
Rule : wait with interrupt Shell banner:
Welcome to Diagnostic Mode

Wait banner :
Waiting for IOS Process

Method : ssh Rule : wait Shell banner: Wait banner :
```

```
Method : console
Rule : wait with interrupt Shell banner:
Wait banner :
```

Using the factory reset Commands

The **factory reset** commands are used to remove all the customer specific data on a router/switch that has been added. The data can be configuration, log files, boot variables, core files, and so on.

The **factory-reset all** command erases the bootflash, nvram, rommon variables, licenses, and logs.

```
Router#factory-reset all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
*Enter*

*May 12 09:55:45.831: %SYS-5-RELOAD: Reload requested by Exec. Reload Reason: Factory Reset.

***Return to ROMMON Prompt
```

Support for Security-Enhanced Linux (SELinux)

Security-Enhanced Linux is a set of patches to the Linux kernel and some utilities to incorporate a strong, flexible mandatory access control (MAC) architecture into the major subsystems of the kernel. SELinux provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements, which allows threats of tampering and bypassing of application security mechanisms to be addressed and enables the confinement of damage that can be caused by malicious or flawed applications.

SELinux enforces mandatory access control policies that confine user programs and system servers to the minimum amount of privilege they require to do their jobs. This reduces or eliminates the ability of these programs and daemons to cause harm when compromised (for example, via buffer overflows or mis-configurations). This confinement mechanism operates independently of the traditional Linux access control mechanisms.

The are no additional requirements or configuration steps required to enable or operate the SELinux feature. The solution is enabled/operational by default as part of the base IOS-XE software on supported platforms.

The following are enhanced show commands that have been defined for viewing SELinux related audit logs.

show platform software audit all

show platform software audit summary

show platform software audit switch *<<1-8> | active | standby> <FRU identifier from a drop-down list>*

Command Examples

The following is a sample output of the **show software platform software audit summary** command:

```
Device# show platform software audit summary
===================================
AUDIT LOG ON switch 1
```

```
-----------------------------------
AVC Denial count: 58
===================================
```

The following is a sample output of the **show software platform software audit all** command:

```
Device# show platform software audit all
===================================
AUDIT LOG ON switch 1
-----------------------------------
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438600.896:119): avc:  denied  { execute } for  pid=8300 comm="sh"
name="id" dev="loop0" ino=6982 scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
 tcontext=system_u:object_r:bin_t:s0 tclass=file permissive=1
========== END ============
```

(output omitted for brevity)

The following is a sample output of the **show software platform software audit switch** command:

```
Device# show platform software audit switch active R0
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438624.916:122): avc:  denied  { execute_no_trans } for  pid=8600
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438648.936:123): avc:  denied  { execute_no_trans } for  pid=9307
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438678.649:124): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
```

```
type=AVC msg=audit(1539438696.969:125): avc:  denied  { execute_no_trans } for  pid=10057
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438732.973:126): avc:  denied  { execute_no_trans } for  pid=10858
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438778.008:127): avc:  denied  { execute_no_trans } for  pid=11579
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438800.156:128): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438834.099:129): avc:  denied  { execute_no_trans } for  pid=12451
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438860.907:130): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
========== END ============
===================================
```

Syslog Message Reference

Facility-Severity-Mnemonic

- %SELINUX-3-MISMATCH

Severity-Meaning

- ERROR LEVEL Log

Message Explanation

- A resource access was made by the process for which a resource access policy is not defined. The operation was flagged but not denied.
- The operation continued successfully and was not disrupted. A system log has been generated about the missing policy for resource access by the process as denied operation.

Recommended Action

- Please contact CISCO TAC with the following relevant information as attachments:
  - The message exactly as it appears on the console or in the system log.
  - Output of "show tech-support" (text file)
  - Archive of Btrace files from the box using the following command ("request platform software trace archive target <URL>") For Example:

Device#**request platform software trace archive target flash:selinux_btrace_logs**