Understand Resilient Infrastructure on IOS XE Devices

Contents

Introduction

This document describes the Cisco approach to Resilient Infrastructure, which is rooted in secure-by-default and secure-by-design.

Prerequisites

Requirements

While there are no specific requirements for this document, a basic understanding of Cisco IOS ® XE software is extremely helpful.

Components Used

The information in this document is applicable to all devices that can run Cisco IOS XE 17.18.2 and later software. This includes Cisco IOS XE routers, switches, and WLCs.

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.

Goal

Our goal is to meaningfully decrease the attack surface on Cisco networking products and minimize security vulnerabilities through secure default settings, removal of insecure legacy technologies and features, and enhanced product security.

You can find more details regarding the push at Cisco for improving network security posture in the Resilient Infrastructure documentation as well as the Cisco IOS XE Software Hardening Guide. However, this document primarily focuses on the technical aspects and considerations that result from the phased implementation of these vital security changes.

Phased Approach

To ensure a reduced attack surface and adoption of critical security best practices while minimizing disruption and effort to our customers, Cisco is taking a phased approach to removing insecure features and protocols. Please note that the phasing of insecure configurations is feature- or protocol-specific. One feature can remain in the Warning phase while another feature enters the Restriction phase.

Phase One: Warning

Users receive warnings on the CLI when configuring key insecure features. Our goal is to raise awareness of those insecure configurations so customers can begin planning to migrate to more secure options. Cisco strongly recommends addressing any insecure warning messages immediately. Insecure configurations in

the Warning phase does not trigger or require Insecure Mode.

Cisco IOS XE version 17.18.2 is the first software release to introduce the Warning phase for insecure features.

Phase Two: Restriction

Key insecure features are disabled by default and require explicit user action to enable (through the introduction of Insecure Mode). Existing deployments continue to function, but new installations require intentional enablement of those insecure configurations. Please note that some features on Cisco IOS XE platforms can not have a Restriction phase: they can

simply display warnings for several releases before subsequent removal.

Cisco IOS XE version 26.1.1 is the first software release to introduce the Restriction phase for insecure features.

Phase Three: Removal

Obselete, insecure features are removed entirely. The timing of feature removal varies, depending on user impact as well as adoption. For example, widely adopted features like SNMPv2 are to phase out slower than less commonly used ones.

Cisco IOS XE version 26.2.1 is the first software release to introduce the Removal phase for insecure features.

Key Commands

These commands are extremely useful as customers implement more resilient infrastructure. These commands are referenced throughout this document.

- **show system insecure configuration**
  - This command is used to display the currently applied, insecure configurations that are in the Restriction phase. It does not display insecure configurations that are in the Warning phase or Removal phase. This command also displays the time remaining for the next insecure configuration scan (detailed in the Timers and Insecure Configuration Scans section).
- **show system security mode**
  - This command provides a brief output showing whether the device is in Secure Mode or Insecure Mode.
- **show running-config all | include system mode insecure**
  - This command displays the running configuration (including default configurations), filtered on

the system mode insecure keywords. Please refer to the Change Security Mode section or additional details.

- **test system secure all**
  - This command immediately runs an insecure configuration scan and displays the **show system insecure configuration** output. This is helpful to refresh the insecure-flagged configurations after a change without waiting for the scan timer to expire.
- **show system insecure profile**
  - This command displays Restriction-phase insecure configurations that the system is designed to detect on that version of software. The list of insecure configurations in the profile is updated over time as security best practices continue to evolve. This is not reflective of the insecure features currently configured on the device. It is simply a list of all Restriction-phase insecure configurations that the system detects. Please refer to the Hardening Guides in the Additional Resources section for all best security practices.

Caveats and Considerations

Timers and Insecure Configuration Scans

The insecure configuration checks and warning messages detailed throughout this document are scheduled on timers to rate-limit how often they run. When an insecure configuration is corrected, it does not immediately disappear from the **show system insecure configuration** output. There is a delay of up to 30 minutes as the configuration scanner operates on a 30-minute cycle. Likewise, there can be up to a two-minute delay between applying an insecure configuration and its corresponding %SYS-4-INSECURE_CONFIG syslog.

Users can view the time remaining until the next scan runs with the **show system insecure configuration** command. The timer is displayed in the first section of outputs. This first example shows that configuration changes have been made, and the next scan for insecure configurations occur in 8 minutes:

```
<#root>

Device#

show system insecure configuration

============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
Generated: Active Configuration Analysis
Total Active Insecure Commands: 1
Database Type: Active (Current State)
Scan Status: Complete
Next Update:

Pending in 8 min 0 sec <<<----------

Database State: Update Scheduled
============================================================
<snip>
```

This next example shows that no configuration changes have been detected since the last scan, so no additional checks for insecure configurations are needed:

```
<#root>

Device#

show system insecure configuration


============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
Generated: Active Configuration Analysis
Total Active Insecure Commands: 1
Database Type: Active (Current State)
Scan Status: Complete
Next Update:

No pending updates <<<----------

Database State: Stable
============================================================
<snip>
```

Users can force an immediate rescan using the **test system secure all** command. In addition to prompting an immediate rescan, this command displays the **show system insecure configuration** output. This is helpful to refresh the insecure-flagged configurations after a change without waiting for the scan timer to expire.

Insecure Configuration Warnings

Starting in 17.18.2 with the introduction of the Warning phase, users can see this syslog syntax:

```
%SYS-4-INSECURE_CONFIG: Module: <MODULE> - Command: <COMMAND> - Reason: <REASON> - Remediation: <REMEDI
%SYS-4-INSECURE_DYNAMIC_WARNING: Module: <MODULE> - Command: <COMMAND> - Reason: <REASON> - Remediation
```

These message include:

- **Module**: The component that generated the log message (such as LOGGING, HTTP, or LINE)
- **Command**: The specific configuration that triggered the warning message
- **Reason**: The reason why this configuration is flagged as insecure
- **Remediation**: Action needed to migrate to a more secure alternative

These warning messages do not impact service or functionality on the device. The intent is to draw attention to these insecure configurations so they can be proactively mitigated by the user.

Note: Starting in Cisco IOS XE version 26.1.1, the INSECURE_DYNAMIC_WARNING messages indicate insecure configurations in the Warning phase while the INSECURE_CONFIG messages indicate insecure configurations in the Restriction phase. Only Restriction-phase configurations appear in the show system insecure configuration output.

Please note that these logs are seen at boot up or after applying an insecure configuration. In addition, they can reappear on the device periodically. You can find additional details regarding these messages and their syntax in the Resilient Infrastructure Cisco IOS XE Security Warnings Reference.

Example Syslog Seen Shortly After Configuration

These are example syslog messages seen shortly after applying an insecure configuration. As noted in the Timers and Insecure Configuration Scans section, these messages can take up to two minutes to appear after applying the insecure configuration:

```
! Feature in the Warning phase:
    *Jan 1 01:23:45.678: %SYS-4-INSECURE_DYNAMIC_WARNING: Module: HTTP - Command: ip http server - Reas

! Feature in the Restriction phase:
    *Jan 1 01:23:45.678: %SYS-4-INSECURE_CONFIG: Module: FTP - Command: ip ftp source-interface Gigabit
```

Example Syslog Seen on Bootup

These are example messages displayed on bootup. A message is displayed for each insecure configuration the system detects:

```
! Feature in the Warning phase:
    INSECURE DYNAMIC WARNING - Module: HTTP, Command: ip http server , Reason: Legacy protocol poses da

! Feature in the Restriction phase:
    SECURITY WARNING - Module: FTP, Command: ip ftp source-interface GigabitEthernet0/0/0 , Reason: No
```

Insecure Mode

Insecure Mode is introduced starting Cisco IOS XE version 26.1.1. Insecure Mode exists to help bridge the gap between existing, insecure deployments and future, hardened networks. The addition of the Insecure Mode configuration allows customers to continue to operate with existing, insecure features while flagging which configurations pose a security risk and need to be mitigated. It also acts as an acknowledgment of insecure features before attempting to apply them on a factory-default device. Insecure Mode also allows for End-of-Life planning for deprecated features before Phase Three where they are completely removed. The goal of Insecure Mode is to migrate customers to secure-by-design networks while minimizing any

potential disruptions to functionality.

For brand new deployments and fresh installations that are factory default, Secure Mode is set by default (**no system mode insecure**), meaning the device does not allow users to apply Restriction-phase insecure configurations. Users need to explicitly enable Insecure Mode with the **system mode insecure** global configuration in order to apply Restriction-phase insecure features and protocols. Insecure features and protocols in the Warning phase can still be applied in Secure Mode, but they do generate warning messages.

Check Current Security Mode

Users can check whether the device is in Secure Mode or Insecure Mode using the **show system security mode** command. The **show running-config all | include system mode** command also reflects whether the device is in Secure Mode or Insecure Mode. The **all** keyword tells the device to include default configurations in the output, as Secure Mode is the default setting on fresh deployments.

These outputs reflect a device in Secure Mode:

```
<#root>

Device#

show system security mode


System Security Mode :

Secure



Device#

show running-config all | include system mode



no system mode insecure
```

The same commands can be used to check if the device is in Insecure Mode:

```
<#root>

Device#

show system security mode
```

```
System Security Mode :
```

Insecure

```
Device#
```

show running-config all | include system mode

system mode insecure

Change Security Mode

Enable Insecure Mode

Users can enable Insecure Mode with the **system mode insecure** global configuration:

<#root>

```
Device# configure terminal
Device(config)#
```

system mode insecure

Enable Secure Mode

Users can enable Secure Mode with the **no system mode insecure** global configuration:

<#root>

```
Device# configure terminal
Device(config)#
```

no system mode insecure

Requirements to Enable Secure Mode

In order to move to Secure Mode:

- any insecure configuration scanning must be complete, and
- all insecure configurations must be removed from the device

If insecure configuration scanning is not complete, the system prompts the user to try again after the scanning timer has expired:

<#root>

```
Device# configure terminal
Device(config)# no system mode insecure
System secure mode cannot be changed to secure as

insecure configuration scanning is in progress. Try after 4 min 0 sec.
```

Users can force an immediate rescan using the **test system secure all** command.

If, after the timer expires and the configuration scan is complete, the system still detects any insecure configurations, the system does not go into Secure Mode. Those insecure configurations must be removed before the system can enter Secure Mode:

<#root>

```
Device(config)# no system mode insecure
System secure mode cannot be changed to secure as

insecure cli(s) are present in system.
```

Once both of these requirements are met, users can enable Secure Mode:

<#root>

```
Device# configure terminal
Device(config)#

no system mode insecure
%SYS-4-SYSTEM_SECURITY_MODE_CHANGE: System Security Mode Changed from INSECURE to SECURE
```

Apply Insecure Configurations

In Secure Mode, if a user tries to apply a Restricted-phase insecure configuration, an error message is displayed and the configuration is not applied. For example:

<#root>

```
Device# configure terminal
Device(config)# ip ftp source-interface Gi0/0/0


%Error:Insecure configurations are not permitted in secure mode.


 To proceed, set the system mode to insecure using the command


system mode insecure


, and then try again.
Module: FTP, Command: ip ftp source-interface GigabitEthernet0/0/0 , Reason: No encryption is configure


%ERROR: Security policy check failed, configuration can't be applied


Device(config)#end
```

The messages displayed immediately after the configuration attempt note the device is in Secure Mode so the insecure configurations provided cannot be applied. You can confirm that the insecure configurations were not applied:

```
Device# show running-config | include ip ftp source-interface
Device#
```

In order to apply Restriction-phase insecure configurations, users need to explicitly enable Insecure Mode first with the **system mode insecure** global configuration:

<#root>

```
Device# configure terminal
Device(config)#


system mode insecure


Device(config)# end


Device#show running-config all | include system mode


system mode insecure
```

Once the device is in Insecure Mode, the Restriction-phase insecure configurations can be applied. A similar security warning message is displayed upon configuration; however, the insecure configuration is applied:

<#root>

```
Device# configure terminal
Device(config)# ip ftp source-interface Gi0/0/0
```

SECURITY WARNING

```
 - Module: FTP, Command: ip ftp source-interface GigabitEthernet0/0/0 , Reason: No encryption is config
Device(config)# end
Device# show running-config | include ip ftp source-interface
ip ftp source-interface GigabitEthernet0/0/0
Device#
```

Users also see a warning message calling attention to the insecure configuration. Because of timers queuing these messages in order to rate-limit them, this syslog can take up to two minutes to appear after configuration:

```
%SYS-4-INSECURE_CONFIG: Module: FTP - Command: ip ftp source-interface GigabitEthernet0/0/0 - Reason: N
```

Please note that only features and protocols in the Restriction phase requires or triggers Insecure Mode. Features and protocols that are in the Warning phase can still be applied in Secure Mode

Automatic Transition to Insecure Mode

When a Cisco IOS XE device is upgraded to 26.1.1 or later, the system detects any Restriction-phase insecure configurations during the boot process and automatically transitions the device to Insecure Mode. Users do not need to worry about manually adding the **system mode insecure** global configuration themselves, and there is no impact to insecure features when moving to the Restriction phase.

This example walks through the automatic transition to Insecure Mode during the upgrade from 17.18.2 (where there is no Insecure Mode context) to 26.1.1 (which has an explicit Insecure Mode context). The device starts with the insecure **ip ftp source-interface GigabitEthernet0/0/0** configuration applied.

Initially, this device starts on Cisco IOS XE version 17.18.2:

```
Device# show version | include Cisco IOS XE Software
Cisco IOS XE Software, Version 17.18.02
```

There is one insecure configuration detected:

<#root>

```
Device# show system insecure configuration
============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
Generated: Active Configuration Analysis

Total Active Insecure Commands: 1 <<<----------

<snip>

+------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [1/1]
+------------------------------------------------------------
|               Module: FTP
|       Parent Command: NA
|          CLI Command:

ip ftp source-interface GigabitEthernet0/0/0 <<<----------

|          Description: FTP service enabled - transmits credentials and data in plaintext, vulnerable t
|               Reason: No encryption is configured
|          Remediation: Transition to secure file transfer methods using SCP, SFTP, HTTPS protocols
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+------------------------------------------------------------
<snip>

============================================================
                    DATABASE SUMMARY
============================================================
Total Active Entries Processed: 1
<snip>
```

Additionally, there is no concept of Secure Mode or Insecure Mode on this version:

```
Device# show running-config all | include system mode
Device#
```

The device is then upgraded to 26.1.1, which introduces the Secure and Insecure Modes.

```
Device# show version | include Cisco IOS XE Software
Cisco IOS XE Software, Version 26.01.01
```

There is still the same insecure configuration applied:

<#root>

```
Device# show system insecure configuration
```

```
============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
Generated: Active Configuration Analysis

Total Active Insecure Commands: 1 <<<----------

<snip>

+------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [1/1]
+------------------------------------------------------------
|               Module: FTP
|       Parent Command: NA
|          CLI Command:

ip ftp source-interface GigabitEthernet0/0/0 <<<----------

|          Description: FTP service enabled - transmits credentials and data in plaintext, vulnerable t
|               Reason: No encryption is configured
|          Remediation: Transition to secure file transfer methods using SCP, SFTP, HTTPS protocols
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+------------------------------------------------------------
<snip>

============================================================
                    DATABASE SUMMARY
============================================================
Total Active Entries Processed: 1
<snip>
```

Because of the presence of this (or any) Restriction-phase insecure configuration, the system detects and automatically transitions to Insecure Mode:

<#root>

```
Device# show system security mode
System Security Mode :

Insecure
```

And the **system mode insecure** configuration is applied automatically:

<#root>

```
Device# show running-config all | include system mode

system mode insecure <<<----------
```

```
system mode warning periodicity 24
Device#
```

Please note that the presence of Warning-phase insecure configurations does not trigger a transition to Insecure Mode. Only the presence of Restriction-phase insecure configurations trigger the automatic transition.

Hardening Devices

You are strongly encouraged to make every effort to migrate away from insecure features and protocols to more secure methods before the Removal phase (Phase Three). Cisco has integrated some serviceability enhancements to make identifying insecure configurations and correcting them significantly easier.

Identify Insecure Configurations Applied

Users can view Restriction-phase insecure configurations that are currently applied with the **show system insecure configuration** EXEC command. This command is automatically included in the **show tech-support** output in versions 26.1.1 and later. This is an example output from a device with three Restriction-phase insecure configurations applied:

<#root>

Device#

show system insecure configuration

```
============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
Generated: Active Configuration Analysis
Total Active Insecure Commands:
```

3 <<<---------- Number of insecure configurations identified

```
Database Type: Active (Current State)
Scan Status: Complete
Next Update: Pending in
```

10 min 0 sec <<<------- Time remaining until this output refreshes to reflect

```
Database State: Update Scheduled
```

any configuration changes applied.

```
============================================================
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processing 3 active insecure CLI entries
```

```
+------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [1/3]
+------------------------------------------------------------
|
```

Module

```
: FTP
|       Parent Command: NA
|
```

CLI Command

```
: ip ftp source-interface GigabitEthernet0/0/0
|
```

Description

```
: FTP service enabled - transmits credentials and data in plaintext, vulnerable to interception
|
```

Reason

```
: No encryption is configured
|
```

Remediation

```
: Transition to secure file transfer methods using SCP, SFTP, HTTPS protocols
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+------------------------------------------------------------
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processed entry 1: ip ftp source-interface GigabitEtherne
```

<snipped other insecure configurations>

```
============================================================
                    DATABASE SUMMARY
============================================================
Total Active Entries Processed: 3
<snip>
```

This output includes key information regarding the module containing the insecure feature, the parent command or configuration if this is a nested configuration, the specific CLI command that was flagged, the reason it was marked insecure, and the remediation action necessary to correct it.

Users can also view a comprehensive list of all insecure CLI patterns using the command **show system insecure profile**. While **show system insecure configuration** shows Restriction-phase insecure configurations that are currently applied, **show system insecure profile** displays all Restriction-phase insecure configurations that the system is designed to detect. The list of insecure configurations in the

profile is updated over time as security best practices continue to evolve.

Example Remediations for Common Insecure Configurations

These examples demonstrate how users can detect, identify, and remedy several commonly-encountered insecure configurations. Cisco has implemented software to help make identification and mitigation as effortless as possible, whether users leverage the INSECURE_CONFIG syslog messages or the **show system insecure configuration** output.

Insecure File Transfer Method

These are the warning messages seen on the device:

```
%SYS-4-INSECURE_CONFIG: Module: FTP - Command: ip ftp source-interface GigabitEthernet0/0/0 - Reason: N
%SYS-4-INSECURE_CONFIG: Module: FTP - Command: ip ftp username cisco - Reason: No encryption is configu
%SYS-4-INSECURE_CONFIG: Module: FTP - Command: ip ftp password * - Reason: No encryption is configured
```

You can run **show system insecure configuration** to see additional information about these insecure configurations:

<#root>

```
Device#
```

show system insecure configuration

```
============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
Generated: Active Configuration Analysis
Total Active Insecure Commands: 3
Database Type: Active (Current State)
Scan Status: Complete
Next Update: No pending updates
Database State: Stable
============================================================
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processing 3 active insecure CLI entries

+------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [1/3]
+------------------------------------------------------------
|               Module: FTP
|       Parent Command: NA
|          CLI Command:
```

ip ftp source-interface GigabitEthernet0/0/0

```
|          Description: FTP service enabled - transmits credentials and data in plaintext, vulnerable t
```

```
|               Reason: No encryption is configured
|          Remediation: Transition to secure file transfer methods using SCP, SFTP, HTTPS protocols
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+-------------------------------------------------------------
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processed entry 1: ip ftp source-interface GigabitEtherne

+-------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [2/3]
+-------------------------------------------------------------
|               Module: FTP
|       Parent Command: NA
|          CLI Command:
```

ip ftp username <USERNAME>

```
|          Description: FTP service enabled - transmits credentials and data in plaintext, vulnerable t
|               Reason: No encryption is configured
|          Remediation: Transition to secure file transfer methods using SCP, SFTP, HTTPS protocols
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+-------------------------------------------------------------
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processed entry 2: ip ftp username cisco

+-------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [3/3]
+-------------------------------------------------------------
|               Module: FTP
|       Parent Command: NA
|          CLI Command:
```

ip ftp password <PASSWORD>

```
|          Description: FTP service enabled - transmits credentials and data in plaintext, vulnerable t
|               Reason: No encryption is configured
|          Remediation: Transition to secure file transfer methods using SCP, SFTP, HTTPS protocols
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+-------------------------------------------------------------
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processed entry 3: ip ftp password cisco

=============================================================
                    DATABASE SUMMARY
=============================================================
Total Active Entries Processed: 3
<snip>
Device#
```

These logs map directly to these configurations:

```
Device# show running-config | include ip ftp
ip ftp source-interface GigabitEthernet0/0/0
ip ftp username cisco
```

```
ip ftp password cisco
```

Users can mitigate the insecure configurations with these changes:

```
<#root>
Device#
configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Device# (config)#
no ip ftp source-interface GigabitEthernet0/0/0

Device# (config)#
no ip ftp username <USERNAME>

Device# (config)#
no ip ftp password <PASSWORD>
```

Insecure, Legacy SNMP Protocols

This is the warning message seen on the device:

```
%SYS-4-INSECURE_CONFIG: Module: SNMP - Command: snmp-server community * ro - Reason: Legacy protocol po
```

You can run **show system insecure configuration** to see additional information about the insecure configuration:

```
<#root>
Device#
show system insecure configuration

============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================
```

```
Generated: Active Configuration Analysis
Total Active Insecure Commands: 1
Database Type: Active (Current State)
Scan Status: Complete
Next Update: No pending updates
Database State: Stable
============================================================
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processing 1 active insecure CLI entries

+------------------------------------------------------------
| ACTIVE INSECURE CONFIGURATION ENTRY [1/1]
+------------------------------------------------------------
|               Module: SNMP
|       Parent Command: NA
|          CLI Command:
```

snmp-server community <STRING> RO

```
|          Description: SNMP Community string configured - uses insecure SNMPv1/v2c protocol vulnerable
|               Reason: Legacy protocol poses data confidentiality and integrity risks due to lack of e
|          Remediation: Configure SNMP v3 User
|          Config Mode: configure
|               Status: ACTIVE
|             Severity: HIGH
+------------------------------------------------------------
SECURE_CONFIG_ACTIVE_INSECURE_CONFIG_DB_WALK: Processed entry 1: snmp-server community cisco RO

============================================================
                    DATABASE SUMMARY
============================================================
Total Active Entries Processed: 1
<snip>

Device#
```

These logs map directly to this configuration:

<#root>

```
Device# show running-config | include snmp-server
```

snmp-server community <STRING> RO

Customers can remedy this using SNMPv3 with authentication and encryption (authPriv).

Frequently Asked Questions (FAQ)

Q: Why is Cisco making these changes?

A: Cisco is making these changes to enhance the security and resilience of its network infrastructure by

disabling insecure legacy features, introducing stronger protections and monitoring, and simplifying secure operations. These efforts help protect customers against evolving cyber threats, reduce downtime, and prepare networks for future challenges like quantum computing. Overall, the initiative aims to build a modern, secure, and reliable foundation for current and future technologies

Q: What happens when a device with an insecure configuration is upgraded to a release in the Restriction phase for that feature?

A: When a device is upgraded to a Restriction (Phase Two) release for a given feature, the system detects the insecure configurations during the boot process and automatically transitions the device to Insecure Mode.

Q: What happens when a device with an insecure configuration is upgraded to a release in the Removal phase for that feature?

A: When a device is upgraded to a Removal (Phase Three) release for a given feature, removed configurations are no longer available. Users must adhere to standard migration procedures for managing obsolete commands.

Q: Are all insecure features removed in the same release?

A: Not all insecure features are removed in the same release. Cisco adheres to a phased approach to deprecate insecure features in three stages: first issuing warnings when insecure features are configured or detected, then restricting their use by disabling them by default or requiring explicit administrator action (through the introduction of Insecure Mode), and finally removing the features entirely in future releases. Some features can skip the Restriction phase and move directly from Warnings to Removal. The timing of removal varies by feature and platform, with release numbers for warnings, restrictions, and removals differing across operating systems such as Cisco IOS XE, Cisco IOS XR, Cisco NXOS, Cisco ISE, and Cisco ASA/FTD. This staged process ensures minimal disruption and allows customers time to transition to secure alternatives.

Q: When does my insecure feature move into the Restriction or Removal phase?

A: The timing for when your insecure feature moves into the Restriction or Removal phase varies by feature and operating system. For detailed information, please refer to the Feature Deprecation and Removal Details documentation.

Q: What alternatives exist for my particular insecure feature?

A: Customers can refer to the Feature Removal and Suggested Alternatives documentation to identify recommended alternatives to various insecure features and protocols.

Q: How can I see which insecure configurations I currently have applied?

A: To see which Restriction-phase insecure configurations you currently have applied, you can use the

command **show system insecure configuration** on Cisco IOS XE 26.1.1 and later releases. This command provides a comprehensive list of Restriction-phase insecure features configured on the device. Additionally, in Cisco SD-WAN Manager, you can navigate to **Monitor** > **Advisories** and select the **Insecure Configurations** tab to view insecure configurations across devices, configuration groups, and templates, with links to remediation steps. This view is refreshed approximately every 30 minutes to ensure up-to-date information.

Q: How can I see a list of all possible insecure configurations on a given software version?

A: You can use the command **show system insecure profile** to view a complete list of all Restriction-phase insecure CLI patterns that the system is designed to detect. Unlike **show system insecure configuration**, which shows only the insecure configurations currently applied, the profile output includes all known insecure configurations in the Restriction phase and is updated over time as security best practices evolve.

Q: I corrected an insecure configuration. Why does it still show up in the **show system insecure configuration** output?

A: The scan for insecure configurations only runs periodically while in Insecure Mode. This means that after correcting an insecure configuration, the system can not immediately reflect the change until the next scheduled scan occurs, which happens on a 30-minute interval. This scheduling ensures that the latest insecure configuration details are updated and displayed regularly while minimizing the overhead needed to perform the scan. You can use the **test system secure all** command to force an immediate rescan so you do not have to wait for the scan timer to expire.

Q: How can I proactively check which insecure configurations I have applied before upgrading?

A: To proactively check which insecure configurations you have applied before upgrading, prior to Cisco IOS XE 17.18.2, customers can use the Cisco AI Assistant for Support bot available on the Cisco Resilient Infrastructure page, which allows uploading configurations to identify insecure features. A similar tool, the Cisco Config Resilient Infrastructure Tester is another option for customers. Starting with Cisco IOS XE 17.18.2 and later, customers can still use these tools, but you also have the option to directly run the command **show system insecure configuration** on your devices to view the insecure configurations currently applied. However, using the AI Assistant for Support bot and Resilient Infrastructure Tester provide additional AI-driven augmentation beyond the direct CLI command.

Additional Resources

Customers are encouraged to read through this documentation to supplement understanding of security best practices and alternatives to their existing, insecure configurations.

Cisco Resilient Infrastructure - Provides essential background on the transition to enhanced security posture across Cisco devices and users can leverage the Cisco AI Assistant for Support Bot in the bottom-right corner of this page to step through a guided workflow to identify insecure configurations from various outputs

Cisco Config Resilient Infrastructure Tester - A tool that can be used to check for insecure configurations based on a provided running-config

Cisco IOS XE Software Hardening Guide - Details best practices to harden your Cisco IOS XE devices and increase the overall security of your network

Feature Removal and Suggested Alternatives - Documents the list of insecure features and protocols that are planned for eventual removal as well as the recommended alternatives

Feature Deprecation and Removal Details - Documents when specific insecure features and protocols enterr Warning and/or Restriction phases based on the Cisco IOS XE software version

SD-WAN Monitor and Maintain Guide - Insecure Configuration Management Chapter - Covers centralized visibility and actionable remediation for insecure feature configurations in Cisco Catalyst SD-WAN, helping administrators identify and fix vulnerabilities to strengthen network security and maintain compliance

Resilient Infrastructure: Cisco Catalyst SD-WAN and Routing Technical Reference - Security hardening and resiliency playbook for Cisco Catalyst SD-WAN and Routing. It provides prescriptive guidance to identify, remediate, and replace insecure configurations across CLI and UI-based management models, aiming to strengthen security, reduce attack surface, and protect data by transitioning from insecure to secure, resilient alternatives while ensuring consistency across operational models

Cisco C9000 Switching Cisco IOS XE – Resilient Infrastructure Playbook - Focuses on identifying insecure configurations and replacing them with secure, resilient alternatives to strengthen security posture, reduce attack surface, and protect data. The playbook aims to ensure consistency across CLI and UI operational models while enhancing network resiliency and operational simplicity for the Catalyst 9000 family

Cisco 9800 Wireless Resilient Infrastructure - Outlines the phased strategy at Cisco for deprecating insecure features and protocols, providing comprehensive migration paths to secure alternatives to prevent service disruptions during software upgrades. It includes detailed reference tables for affected configurations across line transport, file transfers, and management protocols, alongside guidance on the potential operational impacts of failing to migrate