Cisco C9000 Switching IOS XE – Resilient Infrastructure

Contents

Overview

This document covers a list of commands for the C9000 switching family that have been identified as insecure and gives alternative secure commands that can be leveraged to secure the network while having a similar functionality. The commands are organized into the following sections.

- Line transport
- Device server configurations
- File transfer protocols
- Simple Network Management Protocol (SNMP)
- Miscellaneous
- Passwords

Each section has its own set of commands that have been identified as insecure, together with secure alternatives as well as mitigation steps to take when you upgrade to a later Cisco IOS® XE release that removes support for the insecure commands.

The insecure commands will be restricted and removed in a phased manner over the next several releases, starting in 2026. It is highly recommended, however, that you migrate to the secure alternative commands specified in the sections below to help ensure a secure network as well as smooth upgrades between IOS XE releases.

As of the time of writing this document, the latest extended maintenance train for the C9000 family is IOS XE 17.18.x. The next IOS XE release will be IOS-XE 26.1.x.

With IOS XE Release 17.18.2 and later, we have added support for the Command-Line Interface (CLI) to list all configured insecure commands. Execute the command below to get a list of all insecure commands configured on the switch.

Important: This command is to be treated as an exhaustive list of insecure commands on IOS XE 17.18.2 and later. This document is provided as an overview of the changes that are coming and provides a complete list of the commands that will change over time.

- show system insecure configuration

```
Switch#show system insecure configuration
============================================================
          ACTIVE INSECURE CONFIGURATION DATABASE
============================================================

Generated: Active Configuration Analysis

Total Active Insecure Commands: 2

Database Type: Active (Current State)
Scan Status: Complete
============================================================
```

SECURE _ CONFIG _ ACTIVE _ INSECURE _ CONFIG _ DB _ WALK: Processing 2 active insecure CLI entries

+--

| ACTIVE INSECURE CONFIGURATION ENTRY [1/2]

+--

| Module: HTTP

| Parent Command: NA

| CLI Command: ip http server

| Description: HTTP server enabled - unencrypted protocol vulnerable to eavesdropping and man-in-the-middle attacks

| Reason: Legacy protocol poses data confidentiality and integrity risks due to lack of encryption and authentication

| Remediation: Use http secure server to ensure secure web access

| Config Mode: configure

| Status: ACTIVE

| Severity: HIGH

+--

SECURE _ CONFIG _ ACTIVE _ INSECURE _ CONFIG _ DB _ WALK: Processed entry 1: ip http server

+--

| ACTIVE INSECURE CONFIGURATION ENTRY [2/2]

+--

| Module: AAA

| Parent Command: radius server 192.168.1.1 _ 443 _ 0

| CLI Command: key 7 00051105000A59555B

| Description: RADIUS server key configured with weak encryption (type 0, 7, or plaintext) instead of secure type 6 encryption

| Reason: Configuration employs an Insecure method for password storage

| Remediation: Please consider migrating to a secure alternative such as Type-6

| Config Mode: conf-rad-server

| Status: ACTIVE

| Severity: HIGH

+--

SECURE _ CONFIG _ ACTIVE _ INSECURE _ CONFIG _ DB _ WALK: Processed entry 2: key 7 00051105000A59555B

==========

DATABASE SUMMARY

==========

Total Active Entries Processed: 2

Queue Status: Preserved (read-only traversal)

Memory Status: Allocated and stable

Database Integrity: Verified

==========

SECURITY RECOMMENDATIONS

==========

1. IMMEDIATE ACTION REQUIRED:

   - Review all 2 insecure configurations above

   - Follow remediation steps for each entry

   - Prioritize HIGH severity configurations

2. ONGOING MONITORING:

   - Monitor active configuration changes

   - Implement automated security scanning

   - Regular security configuration audits

3. COMPLIANCE REQUIREMENTS:

   - Document all remediation actions

   - Maintain security configuration baseline

   - Schedule periodic security reviews

==========

Also starting with IOS XE 17.18.2, error messages will display on boot/upgrade for all detected insecure configurations. The format of the generated log will be as follows:

%SYS-4-INSECURE_CONFIG or %SYS-4-INSECURE_DYNAMIC_WARNING.

Both the command output and syslog warnings are typically followed by one or more of the following sections (not all messages include all of the sections):

- **Module:** The IOS XE component that generated the log message, for example, LOGGING, HTTP, or LINE
- **Command:** The specific command configured that triggered the warning message
- **Reason:** The reason why this feature or protocol is insecure
- **Description:** Additional details as to why the feature or protocol is insecure
- **Remediation:** Alternatives or action to take to migrate to a more secure alternative

Example:

```
SECURITY WARNING - Module: SNMP, Command: snmp-server community * * , Reason: Legacy protocol poses data confidentiality and integrity risks due to lack of encryption and authentication, Description: SNMP community string configured - uses insecure SNMPv1/v2c protocol vulnerable to eavesdropping, Remediation: Configure snmp v3 user
```

On IOS-XE 26.1.1 and later, to configure any new insecure command, there is a hard requirement to enable insecure mode first using the command below.

```
Switch (config)# system mode insecure
```

Without enabling the insecure mode, any insecure commands will be rejected.

Note: Cisco advises against the use of insecure mode. This mode will be removed at a later IOS-XE release. Customers are advised to remove any present insecure commands and replace them with secure alternatives as soon as possible.

Note: If you upgrade to IOS-XE 26.1.1 with insecure commands already present in the running-config, then the ‘system mode insecure’ command will be automatically added to the running-config upon upgrade to ensure there is no adverse service impact.

We will now look at each type of command in more detail.

Line transport

Overview

This section covers insecure configurations pertaining to the line transport protocols. The major protocol that is marked insecure is the Telnet protocol. The secure alternative is the Secure Shell (SSH) protocol. Note that as a prerequisite to using SSH, you will need to generate a crypto key on the device. Just enabling the SSH transport without the crypto key would not permit SSH connections to the device.

The line transport protocols considered insecure as of IOS XE 17.18.2 are

- Telnet
- rlogin

What happens if you do not migrate?

If you do not migrate to SSH before upgrading to a later IOS XE release that removes support for all insecure commands, the device will be completely locked out. This occurs since Telnet (and rlogin) commands will no longer be applicable, leaving no transport protocol configured.

If you are in the situation outlined above, to recover the device you will have to physically console into the device and configure SSH as the transport protocol. You will then have to generate a crypto RSA key and enable a username and password. This will enable you to log in remotely to the device. The commands for achieving this are given below.

Commands for restoring remote access to a device via SSH

1. Generate a crypto RSA key.

```
crypto key generate rsa
```

2. Remove existing configurations (line transport configurations and enable SSH).

```
Line #/vty#/console

 Transport input ssh

 Transport output ssh
```

If a username/password is configured, SSH will now be enabled for remote access to the device.

If you are using device consoles to connect to neighboring devices, after SSH is enabled, you can use SSH to log in to the neighboring devices. You will have to ensure that the neighboring devices are also configured to support SSH connections:

ssh -l <USERNAME> <IP _ ADDRESS>

List of affected line transport commands

The following table lists the full set of line transport commands affected by this announcement. If you are running any of these commands, we strongly recommended migrating to SSH as outlined above.

Table 1. Insecure line transport commands

Global Config	line <x/y/z> transport output <rlogin| telnet>
Global Config	line <x/y/z> transport output all
Global Config	line <x/y/z> transport input <rlogin| telnet>
Global Config	line <x/y/z> transport input all
Global Config	line <x/y/z - x/y/z> transport output <rlogin| telnet>
Global Config	line <x/y/z - x/y/z> transport output all
Global Config	line <x/y/z - x/y/z> transport input <rlogin| telnet>
Global Config	line <x/y/z - x/y/z> transport input all
Global Config	line vty 0 n transport output <rlogin| telnet>
Global Config	line vty 0 n transport output all
Global Config	line vty 0 n transport input <rlogin| telnet>
Global Config	line vty 0 n transport input all
Global Config	telnet <IP _ address>

Device server configurations

Overview

This section covers insecure configurations with respect to HTTP. The major protocol that is considered insecure is HTTP. The secure alternative is HTTPS.

The device server protocols considered insecure as of IOS XE 17.18.2 are

- IP HTTP server
- IP BOOTP server

What happens if you do not migrate?

If you do not migrate to HTTPS before upgrading to a later IOS XE release that removes support for all insecure commands, communications over port 80 to the device will not work. This includes the webUI for the device. If you use webUI for configuring the device, you will be locked out of accessing the webUI.

Note that communications over port 80 across the switch (pass-through traffic) will be unaffected. Only traffic over port 80 that is initiated or terminated on the switch in question will be affected.

Additionally, some of the ciphers that can be used over port 443 (HTTPS) are considered insecure, and we recommend migrating to a secure cipher instead.

Command for migrating to HTTPS

We recommend migrating to an HTTPS server (over port 443) instead of port 80. An example configuration is as follows:

```
(config) ip http secure-server
```

List of affected device server commands and commands for configuring a secure cipher

The following table lists the full set of device server commands affected by this announcement. In addition, some ciphers over port 443 are also considered insecure. The table lists commands for specifying secure ciphers.

Table 2. Insecure device server commands and commands for configuring secure ciphers

Global Config mode	ip http server
Global Config mode	ip http tls-version <TLSv1.0/1.1>
Global Config mode	ip http client tls-version <TLSv1.0/1.1>
Global Config mode	ip http secure-ciphersuite ecdhe-rsa-aes-128-cbc-sha
Global Config mode	ip http secure-ciphersuite aes-128-cbc-sha
Global Config mode	ip http secure-ciphersuite aes-256-cbc-sha
Global Config mode	ip http client secure-ciphersuite aes-256-cbc-sha
Global Config mode	ip http client secure-ciphersuite aes-128-cbc-sha

File transfer protocols

Overview

This section covers insecure configurations pertaining to the file transfer protocols. The major protocols that are considered insecure are TFTP and FTP. The secure alternative is the Secure Copy Protocol (SCP). Note that as a prerequisite to using SCP, you will need to enable SSH access on the device. Refer to the Line Transport section above for details on how to do this. Performing an SCP transfer without SSH configured will lead to failure of the transfer.

The file transfer protocols considered insecure as of IOS XE 17.18.2 are

- FTP
- TFTP
- RCP

What happens if you do not migrate?

If you do not migrate to SSH and SCP before upgrading to a later IOS XE release that removes support for all insecure commands, you will be unable to perform file transfer operations using FRP, TFTP, or RCP. This applies to both transfer operations from the switch and operations to the switch.

If you are in the situation described above, you will first need to enable SSH connections on the device (as outlined in the Line Transport section). Once this is done, you will be able to run SCP transfers.

Secure alternative commands

Once SSH is enabled, use the following command to initiate SCP transfers to and from the switch:

`copy scp source: destination:`

Examples

```
Copy a file from a switch to a server (IP address 10.1.1.1):

copy scp bootflash:test _ file username@10.1.1.1

Copy a file from a server (10.1.1.1) to a switch:

copy scp username _ 10.1.1.1:<path-to-file> bootflash:
```

There are several commands that are used to specify connection details. These include specifying source interfaces (or IP addresses) to be used for the file transfers. Most of these can just be included in the SCP command itself and do not require a command to be configured. Other examples include the username and password for the connection. Again, these can just be included in the SCP command syntax itself.

Examples

```
ip rcmp source-interface <>

ip ftp source-interface <>

ip tftp source-interface <>
```

For the above commands, you can use SCP with the VRF simply specified as follows:

copy scp <source> <destination> vrf [vrf-name]

In the case of TFTP connections, the blocksize can be specified with the following command:

```
ip tftp blocksize <>
```

While an alternative is not needed in most cases, you can use the command below to tweak the block size:

```
ip ssh bulk-mode <>
```

List of affected file transfer commands

Table 3. Insecure file transport commands

Exec mode	copy ftp
Global Config mode	ip ftp passive
Global Config mode	ip ftp password <uint8 0..7>
Global Config mode	ip ftp password < uint8 0..7> <string>
Global Config mode	Switch(config) ip ftp source-interface <type> <string>
Global Config mode	ip ftp username <string>
Exec mode	copy <> ftp:
Global Config mode	ip rcmd domain-lookup
Global Config mode	ip rcmd rcp-enable
Global Config mode	ip rcmd rsh-enable
Exec mode	copy <> rcp:
Exec mode	copy rcp: <>
Global Config mode	ip rcmd remote-host
Global Config mode	ip rcmd remote-username
Global Config mode	ip rcmd rsh-disable-command
Global Config mode	ip rcmd source-interface
Global Config mode	ip tftp blocksize <>

Global Config mode	`ip tftp source-interface`
Exec mode	`copy tftp: <>`
Exec mode	`copy <> tftp:`

Simple Network Management Protocol (SNMP)

Overview

This section covers insecure configurations with respect to SNMP. Collecting telemetry from switches is an important aspect of network maintenance and troubleshooting. SNMP itself is a mature solution that can be used to collect a lot of information about different features and processes from the switch. Due to its age, however, SNMP contains a lot of commands that are deemed insecure. The recommendation is to migrate to newer technologies like NETCONF or RESTCONF and using YANG models or streaming telemetry technologies like gNMI or gRPC for richer data collection over a better, more secure transport protocol such as HTTPS.

If, however, you cannot migrate away from SNMP, the recommendation is to use SNMPv3, as it provides robust user-based authentication and message integrity compared with SNMPv1 and v2, which rely on weak, unencrypted community strings. With SNMPv3, the recommendation would be to use a more secure cipher and password type.

What happens if you do not migrate?

If you do not migrate to either NETCONF/RESTCONF with API calls or to SNMPv3 with secure ciphers and passwords, and you upgrade to an IOS XE release that removes support for insecure commands, your SNMP functionality will fail. This means that you will no longer be able to collect information from the switch using SNMP. To recover, you will have to reconfigure your SNMP using SNMPv3 with the recommended ciphers, or you will have to migrate to NETCONF/RESTCONF using API calls instead.

Secure alternative commands

Given the nature of SNMP, it is difficult to provide one-to-one mapping of insecure and alternative commands. The amount of information collected depends on the Object Identifiers (OIDs) that were polled from the switch, and the scope of the commands makes collating them all in this document impossible. A good starting point would be the NETCONF and RESTCONF sections in the programmability guide below.

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1718/b-1718-programmability-cg.html.

If you are migrating to SNMPv3 instead, one advantage is that the OIDs in use will remain the same if you were using SNMP v2 or v2c. The only change will be to the format in which the messages are sent. For this, you will have to enable SNMPv3 using the commands below:

```
snmp-server group <group-name> v3 priv read <view-name> write <view-name>

snmp-server user <username> <group-name> v3 auth sha <auth-password> priv aes 256
<priv-password>

snmp-server host <NMS-IP-Address> traps version 3 priv <priv-password>
```

List of affected SNMP commands

The following table lists the full set of SNMP commands affected by this announcement.

Table 4. Insecure SNMP commands

Global Config mode	`snmp-server user <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <ipv6>`
Global Config mode	`snmp-server user <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <1-99>`
Global Config mode	`snmp-server user <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <std-acl>`
Global Config mode	`snmp-server user <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <ipv6>`
Global Config mode	`snmp-server user <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <1-99 >`
Global Config mode	`snmp-server user <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <std-acl>`
Global Config mode	`snmp-server user <> <> v3 encrypted auth (md5) access <ipv6 | (1-99) | std-acl>`
Global Config mode	`snmp-server user <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <ipv6>`
Global Config mode	`snmp-server user <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <(1-99)>`
Global Config mode	`snmp-server user <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <std-acl>`
Global Config mode	`snmp-server user <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <ipv6>`
Global Config mode	`snmp-server user <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <(1-99) >`

Global Config mode	snmp-server user <> <> v3 encrypted auth (sha/sha2/md5) | (0,6,7) <> priv <3des> | (0,6,7) <> access <std-acl>
Global Config mode	snmp context <> user <> auth sha <> priv aes (128|192|256) <> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global Config mode	snmp context <> user <> auth sha <> priv aes (128|192|256) <> access <ipv6> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global Config mode	snmp context <> user <> auth sha <> priv aes (128|192|256) <> access <std-acl> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global Config mode	snmp context <> user <> auth sha <> priv aes (128|192|256) <> access <1-99> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global Config mode	snmp context <> user <> auth sha <> priv aes (128|192|256) <> access <ipv6> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global Config mode	snmp context <> user <> auth sha <> priv aes (128|192|256) <> access <std-acl> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global Config mode	Snmp-server community <0|7> < > <ro|rw> access <ipv6 | (1-99) | std-acl>
Global Config mode	snmp mib community-map <0|7> <> context <> (engineid | security-name|target-list)
Global Config mode	snmp-server user <> <> v3 auth md5 | (0, 6,7) <> priv <des> | (0,6,7) <>
Global Config mode	snmp-server user <> <> v3 auth md5 | (0, ,7) priv <3des> | (0,6,7) <>

Global Config mode	snmp-server user <> <> v3 auth md5 | (0,7) <> priv <des/3des> | (0,7) <>
Global Config mode	snmp-server user <> <> v3 auth md5 | (0, 6,7) <> priv <des/3des> | (0,6,7) <>
Global Config mode	Snmp-server group <> v3 <auth|noauth> access (ipv6| (1-99) | std-acl)
Global Config mode	Snmp-server group <> v3 priv context <>access (ipv6| (1-99) | std-acl)
Global Config mode	snmp-server host <> version {1|2c} * {0|7} <community>
Global Config mode	snmp-server host <> version {1|2c}* {0|7} <community> udp-port <0-65535>
Global Config mode	snmp-server host <> vrf <1-65535> version {1|2c}* {0|7} <community>
Global Config mode	snmp-server host <> vrf <1-65535> version {1|2c} * {0|7} udp-port <0-65535>
Global Config mode	snmp-server host <> version {3} (auth|noauth) <username>
Global Config mode	snmp-server host <> version (auth|noauth) <community> udp-port <0-65535>
Global Config mode	snmp-server host <> vrf <1-65635> version {3} (auth|noauth) <username>
Global Config mode	snmp-server host <> vrf <1-65635> version {3} (auth|noauth) <username> udp-port <0-65535>
Global Config mode	snmp-server host <> version {1|2c} * {0|7} <community>
Global Config mode	snmp-server host <> version {1|2c} * {0|7} <community> udp-port <0-65535>
Global Config mode	snmp-server host <> vrf <1-65535> version {1|2c} * {0|7} <community> udp-port <0-65535>
Global Config mode	snmp-server host <> vrf <1-65535> version {1|2c} * {0|7} <community>
Global Config mode	snmp context abc user [^]+((credential|access|encrypted))? auth md5 [^]+(access)?
Global Config mode	snmp context abc user [^]+((credential|access|encrypted))? auth sha [^]+ priv des (access)?

Global Config mode	snmp context abc user [^]+((credential|access|encrypted))? auth sha [^]+ priv 3des (access)?
Global Config mode	snmp context abc user [^]+(encrypted))? auth md5 [^]+(access)?
Global Config mode	snmp context abc user [^]+((encrypted))? auth sha [^]+ priv des (access)?
Global Config mode	snmp context abc user [^]+((encrypted))? auth sha [^]+ priv 3des (access)?
Global Config mode	snmp context abc user [^]+((credential|access))? auth md5 [^]+(access)?
Global Config mode	snmp context abc user [^]+ auth sha [^]+ priv des <> access <ipv6>
Global Config mode	snmp context abc user [^]+ auth sha [^]+ priv 3des <> access <1-99>
Global Config mode	snmp context abc user [^]+(encrypted))? auth md5 [^]+(access)?
Global Config mode	snmp context abc user [^]+(encrypted))? auth md5 [^]+ priv des (access)?
Global Config mode	snmp context abc user [^]+((encrypted))? auth md5 [^]+ priv 3des (access)?
Global Config mode	Snmp-server community < > <ro|rw>
Global Config mode	snmp mib community-map <> context <> (engineid | security-name|target-list)
Global Config mode	Snmp-server group <> (v1)
Global Config mode	Snmp-server group <> (v2c)

Miscellaneous

Overview

This section covers insecure configurations that cannot be classified into any of the other sections here. Some examples of the features covered in this section are BOOTP server, Network Time Protocol (NTP) authentication, and logging Transport Layer Security (TLS) profile.

Given the diverse nature of the features covered here, we've collected all of the commands and secure alternatives into a table.

What happens if you do not migrate?

Given the diverse nature of the features in this section, it is difficult to describe a general impact.

If you migrate to a later IOS XE release that removes support for these insecure commands, functionality around the specific features will be impacted.

List of affected miscellaneous commands

Table 5. Miscellaneous insecure commands

Global config mode	`ntp authentication-key <num> md5 <string>`	`Use cipher <> instead`
Global config mode	`logging tls-profile <>` `tls-version TLSv1.1`	`Use TLS 1.2 or later`
Global config mode	`logging tls-profile <>` `ciphersuite <aes-128-cbc-sha | aes-256-cbc-sha>`	`Use cipher <> instead`

Passwords and credentials

Overview

This section covers insecure configurations pertaining to configured passwords and credentials. All types of passwords 0, 5, and 7 passwords are considered insecure, and the recommendation is to use type 6, 8, or 9 passwords instead.

We are taking a different approach with passwords. If you are using passwords of types 0 or 7, the system will automatically attempt to convert that password to a type 6 password. Type 6 passwords use strong AES 128-bit encryption, making them difficult to break. Type 6 passwords have been supported since 2006 onward, and most of the features already support type 6 passwords. However, if there is a feature that doesn't support type 6, insecure mode will be enabled automatically for that feature and deprecation will occur only when type 6 password support is added for that feature.

What happens if you do not migrate?

Ideally, type 0 and 7 passwords should automatically be converted to type 6. Passwords are inherently vital for securing access to the system, and we strongly recommend migrating the device to the newer, more secure alternatives outlined below as soon as possible.

Type 6 password auto-conversion

On upgrade to IOS XE 26.1.1, if a master key is not configured, one will automatically be generated and displayed at boot time. The configured passwords will then be migrated from types 0 and 7 to type 6. The master key is unique per device and is not stored in the configuration file.

However, if a master key was already configured on the box, all auto-converted type 6 passwords will use the same master key.

List of affected password commands

The following table lists the password commands affected by this announcement.

Table 6. Insecure password commands

Global config mode	`enable password [1|7] <password>`
Global config mode	`enable secret [1|7] <password>`
Global config mode	`ip scp password <password>`
Global config mode	`ip dhcp pool <pool _ name> authorization shared-password <password>`
Global config mode	`group-policy server username <username> password [0 7] <password>`
Global config mode	`cts policy-server username <username> password [0 7] <password>`
Global config mode	`cts sxp default password [0 6 7] <password>`
Global config mode	`group-policy server username <username> password [0 7] <password>`
Global config mode	`cts credential id <device-id> password <password>`
Global config mode	`line vty [0 15]` `username <> password <>`
Global config mode	`ip wccp web-cache password`

Closing

The first step of security is to clean up the configurations on the device by removing known insecure commands and migrating to secure alternatives. This document covers most of the commands that are now considered insecure. This must not be treated as an exhaustive list. The first source of truth would be the logs generated on your switches on IOS XE releases 17.18.2 and later.

This document will be updated as more commands are identified as insecure. Please bookmark this document and review it again closer to the release of IOS XE versions 26.2.x and IOS-XE 27.1.x

While efforts are underway to ensure a smooth migration to secure configurations wherever possible, automatic migrations may not be possible for most of the use cases outlined above. The strong recommendation is to migrate to secure alternatives as soon as possible to enable smooth upgrades to later IOS XE releases. Upgrading to IOS XE 26.2.x and IOS XE 27.1.x and later without migrating the commands as described here will lead to issues and is strongly discouraged.

Learn more

To learn more about resilient infrastructure, please visit the following webpage. **https://www.cisco.com/c/en/us/about/trust-center/resilient-infrastructure.html**.

This document is about resilient infrastructure on the C9000 switches. For recommendations on migration, please refer to the IOS XE bulletin.

C96-5763210-00 02/26