Resilient Infrastructure: Cisco IIoT

Overview

This document covers the complete list of commands that have been identified as insecure. To help users improve their infrastructure resiliency, it also presents secure alternative commands that can be leveraged to secure the network while providing similar functionality.

The commands are organized into these sections:

- Line transport
- Passwords
- Device server configurations
- File transfer protocols
- Management Protocols
- Miscellaneous

Each section contains its own set of commands that have been identified as insecure. These sections also provide secure alternatives and mitigation steps to follow when upgrading to later IOS-XE releases that no longer support these commands.

The insecure commands will be restricted and removed in a phased manner over the next several releases, starting in 2026. We strongly recommend migrating to the secure alternatives listed below to ensure a secure network and seamless upgrades between IOS-XE releases.

With IOS-XE 17.18.2 and later, we added support for the CLI to list all configured insecure commands.

Run the **show system insecure configuration** command to get a list of all insecure commands configured on the switch.

```
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, we will display error messages on boot/upgrade for all detected insecure configurations. The generated log will have the format as below.

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

CLI and syslog warnings are typically followed by one or more of the following sections (note that not all messages include every section):

1. **Module**: The IOS XE component that generated the log message, for example LOGGING, HTTP, or LINE
2. **Command**: The specific command configured that triggered the warning message
3. **Reason**: The reason why this feature or protocol is insecure
4. **Description**: Additional details as to why the feature or protocol is insecure
5. **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.

Line transport

This section covers insecure configurations pertaining to the line transport protocols. The major protocol which is marked insecure is the telnet protocol. The secure alternative would be the SSH protocol.

Note: As a prerequisite to use SSH, you need to generate a **crypto key** on the device. Just enabling the **ssh transport** without the **crypto key** will not permit SSH connections to the device.

The transport protocols that are considered insecure as of IOS-XE 17.18.2 are:

- Telnet
- Rlogin

What happens if you do not migrate?

While it is recommended to migrate to SSH as soon as possible, if you upgrade to a later IOS-XE release that removes support for all insecure commands without first migrating to SSH, the device would be completely locked out from remote access.

If you find a similar situation as outlined, then to recover the device you will have to physically console into the device and configure the transport protocol of SSH. You would then have to generate a **crypto rsa key**, enable a **username** and **password** and then be able to login remotely into the device.

The list of secure alternative commands is outlined below.

Secure alternative commands

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
   ```

Once you configure a username and password, SSH would be enabled for remote access to the device.

This also pertains to using device consoles to connect to neighboring devices. After SSH is enabled, you can use SSH to login to the neighboring device. You must ensure the neighboring device is also configured to support SSH connections.

```
ssh -l <USERNAME> <IP_ADDRESS>
```

Table 1. Command list

The table lists the complete set of commands affected by this announcement. If you are currently using any of these commands, it is strongly recommended that you transition to the secure alternative commands.

Command mode	Affected command
Global config	**enable password** [1|7] <password>
Global config	**enable secret** [1|7] <password>
Global config	**ip scp password** <password>
Global config	**ip dhcp pool** <pool_name> **authorization shared-password** <password>
Global config	**group-policy server username** <username> **password** [0 7] <password>
Global config	**cts policy-server username** <username> **password** [0 7] <password>
Global config	**cts sxp default password** [0 6 7] <password>
Global config	**group-policy server username** <username> **password** [0 7] <password>
Global config	**cts credential id** <device-id> **password** <password>

Command mode	Affected command
Global config	**line vty** [0 15] **username** <username> **password** <password>
Global config	**ip wccp web-cache password**

Table 2. IIoT IR related command list

Command mode	Affected command
Global config	**cip security** { **password** <password>| <window timeout value> }
Global config	**lte450 profile id** <id> **authentication** <chap | none | pap> **username** <username> **password** <password>

Device server configurations

This section covers insecure configurations with respect to HTTP configurations. The major protocol which is marked insecure is the HTTP protocol. The secure alternative is the HTTPs protocol.

The complete list of transport protocols marked insecure as of IOS-XE 17.18.2 is:

1. ip http server
2. ip bootp server

What happens if you do not migrate?

If you upgrade to a later IOS-XE that removes support for all insecure commands, communications over port 80 to the device will not work. This includes the web UI for the devices. If you use the web UI to configure the device, you will lock yourself out of the web UI.

Note: Communications over port 80 across the switch (pass-through traffic) remain unaffected. This applies only to traffic over port 80 initiated or terminated on the switch in question. Additionally, some of the ciphers which can be used over port 443 (HTTPs) are also marked as insecure and it is recommended to migrate to a secure cipher instead.

Secure alternative commands

It is recommended to migrate to HTTPs server (over port 443) instead of port 80. An example configuration is provided below.

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

Some ciphers over 443 are also marked insecure. Migrate to a secure cipher as listed.

The complete list of affected commands affected is mentioned in the table.

Table 3. Command list

Command mode	Affected command
Global config	**ip http server**

Command mode	Affected command
Global config	**ip http tls-version** <TLSv1.0/1.1>
Global config	**ip http client tls-version** <TLSv1.0/1.1>
Global config	**ip http secure-ciphersuite ecdhe-rsa-aes-128-cbc-sha**
Global config	**ip http secure-ciphersuite aes-128-cbc-sha**
Global config	**ip http secure-ciphersuite aes-256-cbc-sha**
Global config	**ip http client secure-ciphersuite aes-256-cbc-sha**
Global config	**ip http client secure-ciphersuite aes-128-cbc-sha**

File transfer protocols

This section covers insecure configurations pertaining to the file transfer protocols. The major protocols that are marked insecure are the TFTP and FTP protocols. The secure alternative is to use the SCP protocol. Note that, as a prerequisite to using SCP, you need to enable SSH access on the device. Please refer to the section titled "Line transport" for details on how to enable SSH on the device. Performing an SCP transfer without SSH configured will cause the transfer to fail.

Full list of transport protocols marked insecure as of IOS-XE 17.18.2 are:

- FTP
- TFTP
- RCP

What happens if you do not migrate?

We recommend migrating to the SSH and SCP protocols as soon as possible. If you upgrade to a later IOS-XE version that removes support for insecure commands, the system will prevent you from performing file transfers using FTP, TFTP, or RCP. This restriction applies to both transfers from the switch and transfers to the router.

In this scenario, you must first enable SSH connections on the device (refer to the "Line Transport" section for detailed steps). Once enabled, you can perform SCP transfers.

The following list outlines the necessary commands.

Secure alternative commands

1. Enable **ssh** on the device (see, Line Transport)
2. Initiate **scp** transfers to and from the switch.

```
copy scp <source>: <destination>:
```

Example: 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 server (10.1.1.1) to switch

```
copy scp username_10.1.1.1:<path-to-file> bootflash:
```

There are several commands which 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 **username** and **password** for the connection. Again, these can just be included in the **scp** command syntax itself.

Example:

```
ip rcmp source-interface <>
ip ftp source-interface <>
ip tftp source-interface <>
```

For the given commands, you can use **scp** with the **vrf**.

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

In the case of **tftp** connections, the **blocksize** can be specified with this command.

```
ip tftp blocksize <>
```

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

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

The complete list of impacted commands is given in the table.

Table 4. Command list

Command mode	Affected command
Exec mode	**copy ftp**
Global config	**ip ftp passive**
Global config	**ip ftp password** <uint8 0..7>
Global config	**ip ftp password** <uint8 0..7> <string>
Global config	**ip ftp source-interface** <type> <string>
Global config	**ip ftp username** <string>
Exec mode	**copy** <> **ftp**:
Global config	**ip rcmd domain-lookup**
Global config	**ip rcmd rcp-enable**
Global config	**ip rcmd rsh-enable**
Exec mode	**copy** <> **rcp**:

Command mode	Affected command
Exec mode	**copy rcp**: <>
Global config	**ip rcmd remote-host**
Global config	**ip rcmd remote-username**
Global config	**ip rcmd rsh-disable-command**
Global config	**ip rcmd source-interface**
Global config	**ip tftp blocksize** <>
Global config	**ip tftp source-interface**
Exec mode	**copy tftp**: <>
Exec mode	**copy** <> **tftp**:

Management protocols: SNMP

This section covers insecure configurations with respect to SNMP. Collecting telemetry from switches is an important aspect of network maintenance and troubleshooting. The SNMP protocol itself is a mature solution that can be used to collect a significant amount of information about different features and processes from the switch. Due to its age, there are many SNMP commands that are deemed insecure. The recommendation is to migrate to newer technologies such as NETCONF or RESTCONF (using YANG models), or streaming telemetry technologies such as 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 v2c, which relied on weak, unencrypted community strings. Additionally, with SNMPv3, the recommendation is to use a more secure cipher and password type.

What happens if you do not migrate?

While it is recommended to migrate to NETCONF/RESTCONF, API calls, or SNMPv3 with secure ciphers and passwords, if you upgrade to an IOS-XE release that removes support for insecure commands, your SNMP functionality will fail. This means that information from the switch can no longer be collected using SNMP. To recover, you will have to reconfigure your SNMP using SNMPv3 with the recommended ciphers or migrate to RESTCONF/NETCONF using API calls instead.

Secure alternative commands

Given the nature of SNMP, it is difficult to provide a one-to-one mapping of commands. The amount of information collected depends on the OIDs that were polled from the switch, and the scope of the commands makes it impossible to collate them in this document. A good starting point would be the NETCONF and RESTCONF sections in the programmability guide, see https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1718/b-1718-programmability-cg

If you are migrating to SNMPv3 instead, one advantage is that the OIDs in use will remain the same whether you are using SNMPv1, v2c, or v3. The only change will be to the format in which the messages are sent.

To do this, you will have to enable SNMPv3 using the command as given.

```
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>
```

Table 5. Command list

Command mode	Affected command
Global config	**snmp-server user** <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <ipv6>
Global config	**snmp-server user** <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <1-99>
Global config	**snmp-server user** <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <std-acl>
Global config	**snmp-server user** <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <ipv6>
Global config	**snmp-server user** <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <1-99 >
Global config	**snmp-server user** <> <> v3 auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <std-acl>
Global config	**snmp-server user** <> <> v3 encrypted auth (md5) access <ipv6 | (1-99) | std-acl>
Global config	**snmp-server user** <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <ipv6>
Global config	**snmp-server user** <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <(1-99)>
Global config	**snmp-server user** <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <des> | (0,6,7) <> access <std-acl>
Global config	**snmp-server user** <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <ipv6>
Global config	**snmp-server user** <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <(1-99) >
Global config	**snmp-server user** <> <> v3 encrypted auth (sha/sha2/md5) | (0, 6,7) <> priv <3des> | (0,6,7) <> access <std-acl>
Global config	**snmp context** <> user <> auth sha <> priv aes (128|192|256) <> >router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global config	**snmp context** <> user <> auth sha <> priv aes (128|192|256) <> access <ipv6>

Command mode	Affected command
	>router ospf 1/bgp 1/isis 1 >snmp context ctx community *
Global config	**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	**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	**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	**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	**Snmp-server community** <0|7> < > <ro|rw> access <ipv6 | (1-99) | std-acl>
Global config	**snmp mib community-map** <0|7> <> context <> (engineid | security-name|target-list)
Global config	**snmp-server user** <> <> v3 auth md5 | (0, 6,7) <> priv <des> | (0,6,7) <>
Global config	**snmp-server user** <> <> v3 auth md5 | (0, ,7) priv <3des> | (0,6,7) <>
Global config	**snmp-server user** <> <> v3 auth md5 | (0,7) <> priv <des/3des> | (0,7) <>
Global config	**snmp-server user** <> <> v3 auth md5 | (0, 6,7) <> priv <des/3des> | (0,6,7) <>
Global config	**Snmp-server group** <> v3 <auth|noauth> access (ipv6| (1-99) | std-acl)
Global config	**Snmp-server group** <> v3 priv context <>access (ipv6| (1-99) | std-acl)
Global config	**snmp-server host** <> version {1|2c} * {0|7} <community>
Global config	**snmp-server host** <> version {1|2c}* {0|7} <community> udp-port <0-65535>
Global config	**snmp-server host** <> vrf <1-65535> version {1|2c}* {0|7} <community>
Global config	**snmp-server host** <> vrf <1-65535> version {1|2c} * {0|7} udp-port <0-65535>
Global config	**snmp-server host** <> version {3} (auth|noauth) <username>
Global config	**snmp-server host** <> version (auth|noauth) <community> udp-port <0-65535>
Global config	**snmp-server host** <> vrf <1-65535> version {3} (auth|noauth) <username>
Global config	**snmp-server host** <> vrf <1-65535> version {3} (auth|noauth) <username> udp-port <0-65535>
Global config	**snmp-server host** <> version {1|2c} * {0|7} <community>
Global config	**snmp-server host** <> version {1|2c} * {0|7} <community> udp-port <0-65535>

Command mode	Affected command
Global config	**snmp-server host** <> vrf <1-65535> version {1|2c} * {0|7} <community> udp-port <0-65535>
Global config	**snmp-server host** <> vrf <1-65535> version {1|2c} * {0|7} <community>
Global config	**snmp context abc user** [^]+((credential|access|encrypted))? auth md5 [^]+(access)?
Global config	**snmp context abc user** [^]+((credential|access|encrypted))? auth sha [^]+ priv des (access)?
Global config	**snmp context abc user** [^]+((credential|access|encrypted))? auth sha [^]+ priv 3des (access)?
Global config	**snmp context abc user** [^]+(encrypted))? auth md5 [^]+(access)?
Global config	**snmp context abc user** [^]+((encrypted))? auth sha [^]+ priv des (access)?
Global config	**snmp context abc user** [^]+((encrypted))? auth sha [^]+ priv 3des (access)?
Global config	**snmp context abc user** [^]+((credential|access))? auth md5 [^]+(access)?
Global config	**snmp context abc user** [^]+ auth sha [^]+ priv des <> access <ipv6>
Global config	**snmp context abc user** [^]+ auth sha [^]+ priv 3des <> access <1-99>
Global config	**snmp context abc user** [^]+(encrypted))? auth md5 [^]+(access)?
Global config	**snmp context abc user** [^]+(encrypted))? auth md5 [^]+ priv des (access)?
Global config	**snmp context abc user** [^]+((encrypted))? auth md5 [^]+ priv 3des (access)?
Global config	**snmp-server community** < > <ro|rw>
Global config	**snmp mib community-map** <> context <> (engineid | security-name|target-list)
Global config	**snmp-server group** <> (v1)
Global config	**snmp-server group** <> (v2c)

Miscellaneous

This section covers insecure configurations that cannot be classified into any of the other sections in this document. Some examples of the features covered in this section are the bootp server, ntp authentication, and the logging tls profile.

Given the diverse nature of the features covered here, it is easier to present all commands and their secure alternatives in a tabular format.

What happens if you do not migrate?

It is difficult to describe a general impact here, as the features themselves are diverse.

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

Table 6. Command list

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

Conclusion

The first step toward 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 have been marked as insecure; however, it must not be treated as an exhaustive list. The primary source of truth is the logs generated on your switches running IOS-XE releases 17.18.2 and later.

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

While efforts are underway to ensure a smooth migration to secure configurations wherever possible, automatic migration is unlikely for most of the use cases as outlined. The strong recommendation is to migrate the CLIs to secure alternatives as soon as possible to ensure smooth upgrades to later IOS-XE releases. Failure to migrate the commands before upgrading to IOS-XE 26.2.x, 27.1.x, and later will lead to issues and is strongly discouraged.

To learn more about resilient infrastructure, see https://www.cisco.com/c/en/us/about/trust-center/resilient-infrastructure.html