IOS-XR Security Hardening Guide

Cisco's Guide to Harden Devices Running IOS XR 7.x and beyond
Version 1.0

Cisco Systems, Inc.
http://www.cisco.com

Terminology

All the features mentioned in this guide are applicable to routers running Cisco's IOS XR 7.x and later releases. Wherever there are differences between specific platforms, it will be explicitly called out with a note as shown below.

Platform Specific Note
NCS55xx series routers will not be able to support this feature.
ASR9000 series routers support the feature but with a minor change in the behaviour.

Scope

The primary focus of this document is to provide details of IOS XR security features that can be explicitly enabled by customers to harden the security posture of the devices. This document does not cover the details of the mandatory security features enabled by default like Cisco Secure boot, Chip Guard, IMA (Integrity Measurement Architecture) enforcement, etc. that enable the foundations of trust on the device.

This document provides guidance on what security features can be opted-in based on the needs of a customer and provides only a snippet of configurations where applicable. The detailed configuration guide or feature documentation will have to be referred to for more details on the feature, restrictions, etc. A pointer to the detailed configuration guide or feature documentation will be provided for each of the features.

Wherever applicable the specific release from which a particular feature has been introduced will be mentioned.

Licensing Requirements

Most of the security features mentioned in this document are provided as part of the base IOS XR image and don't need any additional licenses. However, there are a few advanced features that have specific licensing requirements and that will be explicitly mentioned. Wherever there is no mention of licensing, it implies that the base image supports the feature by default.

Introduction

This document contains information that will help users secure Cisco's IOS-XR based routers to increase the overall security posture of the network. This document is structured around the three planes by which the functions of a network device are categorized. The three functional planes of a router namely - the management, control, and data planes - each providing a different functionality that must be protected.

Management Plane: The management plane contains the logical group of all traffic that supports provisioning, maintenance, and monitoring functions for the Cisco IOS XR device and the network. Traffic in this group includes SSH, SCP, Simple Network Management Protocol (SNMP), Syslog, TACACS+ and RADIUS, DNS, NetFlow, and Cisco Discovery Protocol. Management plane traffic is always destined to the local Cisco IOS XR device.

Control Plane: The control plane contains the logical group of all routing, signaling, link-state, and other control protocols that are used to create and maintain the state of the network and interfaces such as Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), Label Distribution Protocol (LDP), Intermediate System to Intermediate System (IS-IS), and Address Resolution Protocol (ARP), and Layer 2 keepalive. Control plane traffic is always destined to the local Cisco IOS XR device.

Data Plane: The data plane contains the logical group of "customer" application traffic generated by hosts, clients, servers, and applications that are sourced from and destined to other similar devices supported by the network. Data plane traffic is mainly forwarded in the fast path and is never destined to the local Cisco IOS XR device.

Secure Operations

Secure network operations are a substantial topic. Although most of this document is devoted to the secure configuration of a Cisco IOS XR device, configurations alone do not completely secure a network. The operational procedures in use on the network, as well as the people who administer the network, contribute as much to security as the configuration of the underlying devices.
The following sections contain operational recommendations that network administrators are advised to implement. These sections highlight specific critical areas of network operations and are not comprehensive.

Monitor Cisco Security Advisories

The Cisco Product Security Incident Response Team (PSIRT) creates and maintains publications, commonly referred to as PSIRT Advisories, for security-related issues in Cisco products. Cisco PSIRT also publishes advisories to communicate less severe issues. Security advisories are available at http://www.cisco.com/go/psirt.

See the Cisco Security Vulnerability Policy for more information about Cisco PSIRT vulnerability reporting.
To maintain a secure network, network administrators should be aware of the information communicated in Cisco Security Advisories. Detailed knowledge of a vulnerability is required prior to evaluating the threat that the vulnerability can pose to a network. Refer to Risk Triage for Security Vulnerability Announcements for assistance with this evaluation process.

Establishing Hardware Integrity

With the increase of counterfeit hardware sold through illegal channels, it becomes critical to validate the authenticity of the hardware before proceeding with other hardening measures. Every Cisco hardware comes with a platform security chip called Trust Anchor module (TAm) chip that helps establish the hardware integrity. Customers can cryptographically challenge the router to provide its unique hardware identity called SUDI (Secure Unique Device Identity). The router responds to the challenge by signing the response with the device unique SUDI private key. Customers can then verify the signature of the response and proceed with further validation of the hardware identity. More details of this workflow are captured in the "Establishing Hardware Integrity using SUDI" section in the Management Plane Hardening part of this guide.

Identity & Access Management

The Authentication, Authorization, and Accounting (AAA) framework is vital to securing network devices. The AAA framework provides authentication of management sessions, limits users to specific, administrator-defined commands, and logs all commands entered by all users.
See the Authentication, Authorization, and Accounting section of this guide for more information about leveraging AAA.

Centralized Log Collection and Monitoring

To gain an understanding of existing, emerging, and historic events that are related to security incidents, an organization should have a unified strategy for event logging and correlation. This strategy must leverage logging from all network devices and use prepackaged and customizable correlation capabilities.
After centralized logging is implemented, a structured approach must be developed to log analysis and incident tracking. Based on the needs of the organization, this approach can range from a simple diligent review of log data to an advanced rule-based analysis.
See the Logging Best Practices section of this guide for more information about how to implement logging on Cisco IOS XR network devices.

Use of Secure Protocols

Many protocols are used to carry sensitive network management data. Secure protocols should be used whenever possible. A secure protocol choice includes the use of SSH instead of Telnet so that both authentication data and management information are encrypted.

See the "Securing Interactive Management Sessions" section of this document for more information about the secure management of Cisco IOS XR devices.

Traffic Visibility

NetFlow provides the ability to monitor traffic flows in the network. Intended to optimally export traffic information to network management applications, NetFlow can also be used to show flow information on a router. This capability allows a network administrator to see what traffic traverses the network in real time depending on the sampling rate configured for NetFlow export.

Configuration Management

Configuration management is a process by which configuration changes are proposed, reviewed, approved, and deployed. In the context of a Cisco IOS XR device configuration, there are configure commit point records for each configuration change. These records can be used to determine what security changes were made and when these changes occurred. In conjunction with AAA log data, this information can assist in the security audit of network devices.
The configuration of a Cisco IOS XR device contains many sensitive details, including usernames, passwords, and the contents of access control lists (ACLs). The repository used to archive Cisco IOS XR device configurations should be secured and access should be restricted to only those roles and functions that require access. Insecure access to this information can undermine the security of the entire network.

Data Protection

Data protection includes data-at-rest protection for sensitive data and safely sanitizing the persistent storage devices before decommissioning or RMAs. Sensitive data like device configuration needs protection at rest. Features like SSD encryption provided by IOS XR must be enabled to encrypt the disk partitions that hold the device configuration.

Secure Onboarding

To ease the onboarding of new devices into the network, features like Zero Touch Provisioning (ZTP) could be used by customers. However, there are a few aspects of trust to be considered as listed below which brings in the need to adopt Secure ZTP instead of the classic ZTP.

1. Trusting the network device being provisioned by the provisioning server.
2. Trusting the provisioning server by the network device.
3. Trusting the artefacts sent by the provisioning server.

To use Secure ZTP, customers can make use of Cisco's MASA service to fetch ownership vouchers of the devices they own and then proceed with the provisioning process. Please refer to the Secure ZTP section of the guide for more details.

Consent Based Security Features

In the context of security hardening, customers must also consider the threats from rogue internal employees too in addition to external threats. To help with countering both external and internal threats, additional consent-based security feature provided by IOS XR can be implemented. The additional consent can be for sensitive features like enabling/disabling Lawful Interception, malicious re-imaging of routers, etc. The "Consent Based Security Features" section of this document has more details on this.

Management Plane

The management plane consists of functions that achieve the management goals of the network. Such goals include interactive management sessions using SSH, as well as statistics gathering with SNMP or NetFlow. When considering the security of a network device, it is critical that the management plane is protected. If a security incident undermines the functions of the management plane, network recovery or stabilization may not be possible. The following sections detail the security features and configurations available in Cisco IOS XR Software that help fortify the management plane.

General Management Plane Hardening

The management plane is used to access, configure, and manage a device, as well as monitor its operations and the network on which it is deployed. The management plane receives and sends traffic for operations of these functions. Both the management plane and control plane of a device must be secured as operations of the control plane directly affect operations of the management plane. The following list includes protocols that are used by the management plane:

- Simple Network Management Protocol (SNMP)
- Telnet
- SSH
- SFTP
- FTP
- TFTP
- Secure Copy Protocol (SCP)
- TACACS+
- RADIUS
- NetFlow (also used by the Data Plane as that is where the traffic comes from)
- Network Time Protocol (NTP)
- Syslog

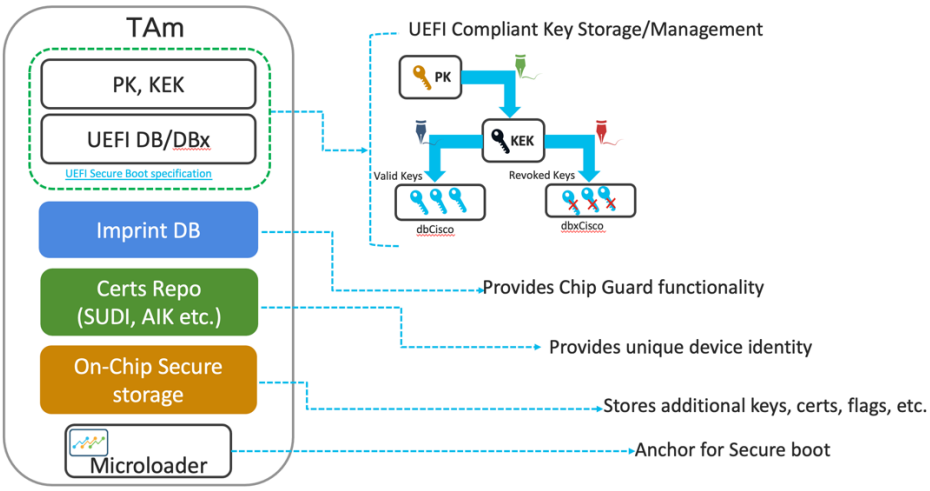
Administrators should take measures to ensure the survival of the management and control planes during security incidents. If one of these planes is successfully exploited, all planes can be compromised.

Establishing Hardware Integrity using SUDI

The first step towards is to ensure the router is a genuine device shipped from Cisco and not counterfeit hardware. This section explains how customers can use Cisco's TAm chip to ensure hardware integrity. Please note that this is an optional step that could be used by customers having the need to validate hardware authenticity.

TAm Chip & SUDI

Cisco's Trust Anchor module (TAm) chip provides an immutable unique hardware identity called SUDI (Secure Unique Device Identity). SUDI enables customers to cryptographically challenge the router to provide its unique identity which could be used to validate the authenticity of the hardware. In addition to SUDI, the TAm chip also has additional functionalities like providing a hardware Root of Trust for Cisco Secure boot, enabling remote attestation through PCR quotes, on-chip secure storage for critical data, securing encryption keys for various features, etc. The various blocks of TAm chip are shown below.



Cisco's Trust Anchor module (TAm) Chip Overview

In case of a modular system, each card (active RP, standby RP, Line cards, Fabric cards, etc.) has its own physical TAm chip and each of the cards can be verified for their hardware authenticity using the card specific SUDI. For a fixed system (pizza box router), there would be one physical TAm chip as they are 1RU systems with a single card.

The SUDI key-pair is unique to each card and the SUDI private key is securely kept on the chip. The SUDI public certificate is available to validate that the router's unique identity is from Cisco and had not been modified.

SUDI Workflow

The SUDI certificate gets programmed inside the TAm chip during manufacturing. The below CLIs can be used by customers to verify the details.

Below is the CLI to query the signed SUDI info from a router.

```
show platform security tam sudi-certs location <location of the card>
```

Below are the steps to verify the SUDI programming inside the TAm chip.

1. SSH to the router and execute the above CLI to get the SUDI certs
2. In case the SUDI certificates are not programmed inside the TAm chip, you will see a CLI output as shown below.

```
RP/0/RP0/CPU0:ios#show platform security tam sudi-certs location
0/RP0/CPU0
Wed Jun 9 10:09:27.266 UTC
-----------------------------------------
Node - node0_RP0_CPU0
-----------------------------------------

Sudi Root Cert:
-----------------------------------------
No Sudi Root Cert

Sudi Sub CA Cert:
-----------------------------------------
No Sudi Sub Ca Cert

Sudi Cert:
-----------------------------------------
No Sudi Cert
RP/0/RP0/CPU0:ios#
```

The CLI output for success case is show below. The below CLI displays the cert in json format and the same can be queried in PEM format too.

```
RP/0/RP0/CPU0:ios#show platform security attest certificate CiscoSUDI
location <location> nonce 1234 json
Mon Mar  1 08:56:25.824 UTC
{"system-certificates":[{"node-
location":"node0_RP0_CPU0","nonce":"EjQ=","certificates":{"certificate":[{"
name":"Cisco SUDI
Root","value":"MIIDITCCAgmgAwIBAgIJAZozWHjOFsHBMA0GCSqGSIb3DQEBCwUAMC0xDjAM
BgNVBAoTBUNpc2NvMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwOTkwIBcNMTYwODA5MjA1ODI
4WhgPMjA5OTA4MDkyMDU4MjhaMC0xDjAMBgNVBAoTBUNpc2NvMRswGQYDVQQDExJDaXNjbyBSb2
90IENBIDIwOTkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDTtuM1fg0+9Gflik4ax
lCK1I2fb3ESCL8+tk8kOXlhfrJ/zlfRbe60xRP0iUGMKWKBj0IvvWFf4AW/nyzCR8ujTt4a11Eb
55SAKXbXYQ7L4YMg+lmZmg/Iv3GJEc3HCYU0BsY8g9LuLMvqwiNmAwM2jWzNq0EPArt/F6RiQKq
6Ta3e7VIfDZ7J65OA2xASA2FrSe9Vj97KpQReDcm6G7cqFH5f+CrdQ4qwAa4zWNyM3kOpUb637D
Nd9m+n6WECyc/IUD+2e+yp21kBZIKH7JvDpu2U7NBPfr52mFX8AfCZgkXV69bp+iYfsaH1DvXIf
PpNp93zGKUSXxEj4w881t2zAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMBAf8EBTAD
AQH/MB0GA1UdDgQWBBQ4lVcPNCNO86EmILoUkcdBiB2jWzANBgkqhkiG9w0BAQsFAAOCAQEAjeK
Zo+4xd05TFtq99nKnWA0J+DmydBOnPMwYlDrKfBKe2wVu5AJMvRjgJIoY/CHVPaCOWH58UTqfji
95eUaryQ/s36RKrBgMMlwrWNItxE625PHuaN6EjD1WdWiRMZ2hy8F4FCKz5hgUEvN+PUNZwsPnp
U6q3Ay0+11T4TriwCV8kJx3cWu0NvTypYCCXMscSfLFQR13bo+1z6XNm30SecmrxkmQBVMqjCZM
```

VvAxhxW1iGnYdPRQuNqt0xITzCSERqg3QVVqYnFJUkNVN6j0dmmMVKZh17HgqLnFPKkmBlNQ9hQ
cNM3CSzVvEAK0CCEo/NJ/xzZ6WX1/f8Df1eXbFg==","name":"Cisco SUDI
CA","value":"MIIEZzCCA0+gAwIBAgIJCmR1UkzYYXxiMA0GCSqGSIb3DQEBCwUAMC0xDjAMBg
NVBAoTBUNpc2NvMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwOTkwIBcNMTYwODExMjAyODA4W
hgPMjA5OTA4MDkyMDU4MjdaMDExHzAdBgNVBAMTFkhpZ2ggQXNzdXJhbmNlIFNVREkgQ0ExDjAM
BgNVBAoTBUNpc2NvMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvdzeSWdDI6lRZDY
RvA6JqaRvQyy6Dx1WaqI82UeKR4ZRn0efxMGvp4c88/VMS8WSjQO1qolMfMxqHkcSiFBOULx6Tr
quw4TrEf9sIuzvgJvDaEa8IllXPwtPtNqZEIWi8jlinz2uGam93KuGPcioHfruzbDKWHL/HWFGY
Mgz+OKwhD3J4NRySknQvUovfV8eWLeVOqW8rbnG3TZxv5VexOiK4jL3ObvsQPuAWUwUoo7nuFlE
GTG/VCeyCe/H8+afIScbZOkI9xejtckflnBYFVCyFxzm2H3YZatb6ohbyRXLtOPjT3SJ+OOoYMl
SLd28z727LpRbFFLGYhyWxEXDuQIDAQABo4IBgjCCAX4wDgYDVR0PAQH/BAQDAgEGMBIGA1UdEw
EB/wQIMAYBAf8CAQAwfwYIKwYBBQUHAQEEczBxMEEGCCsGAQUFBzAChjVodHRwczovL3d3dy5ja
XNjby5jb20vc2VjdXJpdHkvcGtpL2NlcnRzL2NyY2EyMDk5LmNlcjAsBggrBgEFBQcwAYYgaHR0
cDovL3BraWN2cy5jaXNjby5jb20vcGtpL29jc3AwHwYDVR0jBBgwFoAUOJVXDzQjTvOhJiC6FJH
HQYgdo1swUgYDVR0gBEswSTBHBgorBgEEAQkVAR4AMDkwNwYIKwYBBQUHAgEWK2h0dHA6Ly93d3
cuY2lzY28uY29tL3NlY3VyaXR5L3BraS9wb2xpY2llcy8wQwYDVR0fBDwwOjA4oDagNIYyaHR0c
DovL3d3dy5jaXNjby5jb20vc2VjdXJpdHkvcGtpL2NybC9jcmNhMjA5OS5jcmwwHQYDVR0OBBYE
FOpro7nBE5d+G/s6jWhgBzlfh0j6MA0GCSqGSIb3DQEBCwUAA4IBAQBcqYEOgAHhGWKndwM901X
X2Enh4hjXR5avDg7G/f6Tb9H509dtQW+AeZGEghhwUrw1EeG79tHkncAe+m+64xMC1ttyI1RSyn
8rBqQYkXnnCRbtF/NwpQe5fjvdeIFWJhUI16TOt/ZlkNnWnLsUU1alZmN+J/FhSr8VTJWGRM9gY
8hefH8f5U7LMiDXxsFVHB7R6KGNjvtawrl6W6RKp2dceGxEIIvMahgMWWHHiWOQAOtVrHuENEjY
R/7klLLwdgQF/NNCA2z47pSfMFnBcr8779GqVIbBTpOP2E6+1pBrE2jBNNocuBG1fgvh1qtJUdB
bTziAKNoCo4sted6PW2/U"},{"name":"Cisco
SUDI","value":"MIIEXzCCA0egAwIBAgIEApocYzANBgkqhkiG9w0BAQsFADAxMR8wHQYDVQQD
ExZIaWdoIEFzc3VyYW5jZSBTVURJIENBMQ4wDAYDVQQKEwVDaXNjbzAgFw0yMDExMTIwMzI2MTB
aGA8yMDk5MDgwOTIwNTgyNlowgacxIDAeBgNVBAUTF1BJRDo4MjAxIFNOOkZPQzI0MzNLR1gxMQ
4wDAYDVQQKEwVDaXNjbzEYMBYGA1UECxMPQUNULTIgTGl0ZSBTVURJMVkwVwYDVQQDDFBDaXNjb
yA4MjAxIENoYXNzaXMgdy8gMjR4NDAwR0UgUVNGUDU2LUREICYgMTJ4MTAwR0UgUVNGUDI4IE1B
Qzo2NDNhZWFjNzljMDAtMDIwMDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAK6xUIY
NdYCpxmbei2YQjTkPuKtFNssQcjczpln8riz3M1USRX6vXC5FjKDRg7m4XAyzrLoVPPiddWhpFP
p2hwQvtbJpCo6UMfLFCRY2OXna30aDrndGhyMsGVBspiUjpvpPqUeDiDM7CS+wMWcLuWO8qgzPm
EvWlOxP6mGHygZ/C6rDk1W27fdhflcVEi0PNAns3cI/C5jD0mohhYmhajnz4y20s1ulCKkFoXA3
R0oyCXo2ijQqh51plBCNmgN3o8IHtR4sL+i5wbkunDFKwiJlTREYe5LqjPh+tFBh//4RSJsmO8+
tohfUZjuhkLZiRNTqNHAkvwIbhQrVQ0bQo8cCAwEAAaOCAQQwggEAMA4GA1UdDwEB/wQEAwIF4D
AMBgNVHRMBAf8EAjAAMB8GA1UdIwQYMBaAFOpro7nBE5d+G/s6jWhgBzlfh0j6MIGfBgNVHREEg
ZcwgZSgTgYKKwYBBAEJFQMEAhNAMDk0QzBFQkEyMjIwOTE2MjFDMjREOUQ2OEJDOUU1OTJGNzBG
RTdCREI0QjI3MzczREU5QkUxM0NGQ0UxMzFCNaBCBgkrBgEEAQkVAgOgNRMzQ2hpcElEPWFWZFM
4WUtFQ0hHNVB6VGpWTDU4U0FBQUFBQUFBQUFBQUFBQUFBQUE9MB0GA1UdDgQWBBSdvQWyyL
zRyFge2UF+sd7zdSJwTzANBgkqhkiG9w0BAQsFAAOCAQEAt6m9YR+WOEHxOKiPDjPed5hGQH53d
KMp1mKcw7KcctO03QnpK6y0aG0KDcTKGvnM68UWG5Z20b4AcE807kf3I0MZ5DpgUOYKlXmhFF5T
Vfue/YgoKMFtyTqfh+YU8JVS/L8ArY+qwbgeZliWyEQKVns50+jGp5/tTGv51r/Wsyl9B6luXk0
5fmUCJfOrkCMQ4ssfsf5A8C8sxCQSnTzb7GW4ayDsO2runSm+taoY0Q1Dsf/OCWZwE4/P51GLWf
IVLw6EpT063hkraTZBE2peNo3wdSHeQREYqwJznLWoEinM8tXwC+6c1B96hvnkFJwuUkI56z1Uz
JvkL9Jm5wTfEg=="}]},"signature":"GD/ttJor4ErmRj9A+ppANyPIgxW3bHnLzoStJXbgvC
EH/irGsveQfjetqnnMac2ON1QNi8AX683OnAO1E23bHFZ1ARZkd2KwtDgqIOWs4YOYq5Ui9DSjc
iGMKH+YtNHFrRGxEl29k91/hhYSu1Em2hCBRrH1NTSpE/c2SoZHD/suFKOZZJNUuTCN3tbdSHjU
fT5wluMukc3BgFhOkI2sUS8vCWE6bEKePZoAn+EDuX0h2chfXPbmtEVKWGzv8DU+hYgyERo8EtT
JNTfQq8FhDoMaRcjlj7b7wNcMaZByTFl11NtbKYoF2WrOslmgOOsXqT9EJuCwHq+gLjPlgfXi8g
==","signature_version":2}]}

Openssl command can be used to verify the certificate chain and the signature of the response received from the router.

The location to fetch the corresponding root CA and sub-CA are pasted below:
Root CA: https://www.cisco.com/security/pki/certs/crca2099.pem
Subordinate CA: https://www.cisco.com/security/pki/certs/hasudi.pem

As mentioned earlier, the router can be challenged to provide its identity where the response is signed by the router unique SUDI private key. Below CLIs can be used to perform this validation.

Below is the CLI to query the signed SUDI info from a router.

```
show platform security attest certificate CiscoSUDI location all nonce 1234
json
show platform security attest certificate CiscoSUDI location 0/RP0/CPU0
nonce 1234 json
```

Supported releases – This feature is supported from IOS-XR releases 7.2.2 and beyond.

Yang support is available for all the above-mentioned CLIs. Here is the link to the yang model – GitHub Link

Below are the steps to be performed to perform the SUDI validation.

1. SSH to the router and execute the above CLI to get signed SUDI certs
2. Signature validation to be done off-box using the below steps
   a. Cert chain validation – Root, Sub-CA, and leaf certificate
   b. Signature validation with the SUDI leaf cert
   c. Verify the SN & PID info

The scripts needed to perform this SUDI validation to detect counterfeit hardware can be referenced from here - https://github.com/ios-xr/sudi_verifier.

CLI Signature Utility

Description

With signature utility, any IOS-XR CLI output can be signed by SUDI private key to ensure the output is fetched from the intended router and not replayed by a malicious attacker. The signature of the signed output received from the router can be validated using SUDI cert before consuming the CLI output.

This features helps in countering the below threat scenarios even though SSH is used for the management plane connection.

- The connection to the device might be multiple hops away where some of the intermediate hops are not controlled by the device owner.
- Provides additional confirmation that the output from the device can be trusted and is not replayed by a meddler-in-the-middle (MITM). Aligns with the principles of defense-in-depth to have another layer of validation.

This feature is supported in all IOS XR 7.x releases.

Verification Steps

1. SSH to the router and execute the required CLI and pipe it to the signature utility with a unique nonce to get signed CLI output.

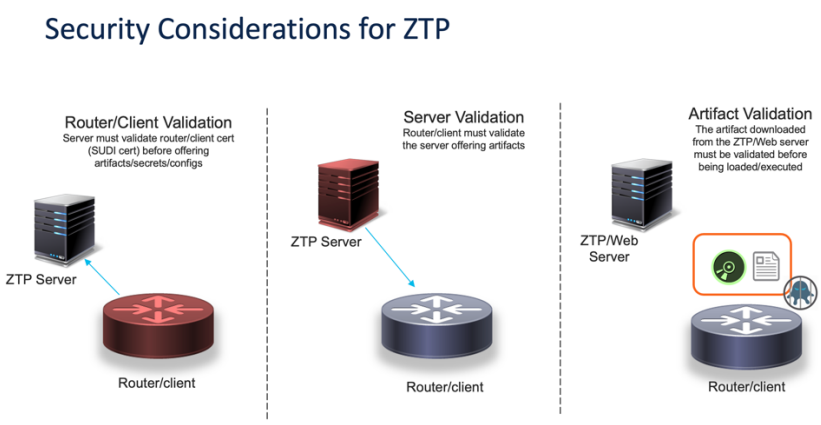
2. Signature validation to be done using SUDI leaf certificate fetched using the CLIs mentioned in the previous section.
3. If signature validation passes, consume the actual CLI output.

Sample output is pasted below. The nonce ensures freshness of the response and is included in the signature.

```
RP/0/RP0/CPU0:Galapagos#show version | utility sign nonce ABCD
Tue Mar  7 06:31:00.071 UTC
{
 "cli-output": "Cisco IOS XR Software, Version 7.4.1 LNT\nCopyright (c)
2013-2021 by Cisco Systems, Inc.\n\nBuild Information:\n Built By     :
ingunawa\n Built On      : Wed Aug 04 08:28:43 UTC 2021\n Build Host   :
iox-lnx-021\n Workspace    : /auto/srcarchive17/prod/7.4.1/ncs540l-
aarch64/ws\n Version      : 7.4.1\n Label        : 7.4.1\n\ncisco
NCS540L\ncisco N540X-8Z16G-SYS-A processor with 8GB of memory\nGalapagos
uptime is 6 weeks, 6 days, 19 hours, 9 minutes\nCisco NCS 540 Series Fixed
Router 12x1G, 4
xCu, 8x1/10G, AC\n\n",
 "signature-envelop": {
  "nonce": "ABCD",
  "signature-version": "02",
  "sudi-signature":
"Ugj00x78n3hXD1nzymEOpUsR143N3Zgz8g15m40eQmrO6yQ0etqGM+10vkSEoz9zTnQ+qicufZ
yp+Vx4MRLagnFXOoQubAY94CB/85qmrLi1is9phjPJ0uDhK5bpF8bQZtZbQ3PcLOyfx1sG8Gk13
I0xQaWdgbB1daz3setsjHkjvHzFSu2aTtKW+DdZSUOxOaCXgSxazwDbE/v826Lng31JzFfgh9SL
QEijp3IfdmKFeRpdK4fOSZN1tXdwlfXRo2YpRPEf9oPEYXI91/b5Bjaz+kCamGintVeqV5XiBxL
vpVLtxIymoZtJuDdX/NYe/5UGtjG/wAMcgbN/1JKlBQ=="
 }
}
```

Secure Zero Touch Provisioning (SZTP)

For zero touch provisioning at scale, customers have been using IOS XR's ZTP feature for quite some time. However, there are some threats to be considered when using classic ZTP as shown below.



The primary threats are,

1. Router validation by the provisioning server
2. Provisioning server validation by the router
3. Artefacts validation by the router

To address the above threats with classic ZTP, customers are strongly encouraged to adopt Secure ZTP which is based on RFC8572. Secure ZTP addresses the above threats using the below features of IOS XR platforms.

1. Router validation – This is addressed using the SUDI certification mentioned in the previous sections. The provision server uses SUDI challenge/response to establish the authenticity of the device before starting the provisioning process.
2. Server validation – Using Ownership vouchers (RFC8366), the problem of server validation can be addressed. Customers can establish ownership of the devices by onboarding their own Pinned Domain Certificate (PDC). Once this is established, the device can then trust the provisioning server to establish a secure connection with it.
3. Artefact validation – The boot artefacts provisioned from the server can be signed by a certificate chained to the same root CA as the PDC. This helps in the device validating the signature of the artefacts before consuming.

Ownership Establishment

Ownership establishment is the process through which customers can onboard their own certificates on to the device. This is based on RFC8366. Once ownership is established, it can be used for features like Secure ZTP, Re-image protection, etc.

Ownership Vouchers & MASA Service

Customers can request Ownership Vouchers through Cisco's MASA service. Details on establishing ownership can be found in the configuration guide here.

For more details on adopting Secure ZTP workflow, please refer to the configuration guide here.

Password Management

Passwords control access to resources or devices, and administrators define passwords to authenticate requests. When a request is received for access to a resource or device, the request is challenged for verification of the password and identity, and access can be granted, denied, or limited based on the result. Security best practices dictate that passwords should be managed with a TACACS+ or RADIUS authentication server. However, a locally configured password for access is still required if TACACS+ or RADIUS services fail.

A device can also have other password information present within its configuration, such as network time protocol (NTP) key, a simple network management protocol community string, or routing protocol key, such as BGP and Message-Digest algorithm 5 (MD5) Authentication.

When it comes to password management, there are two types of methods available. For scenarios where a 2-way encryption/decryption is needed (like key-string for MACsec under key chain), customers can use type-6 encryption. For user login to the devices, a one-way password hashing mechanism is used.

The following 2 sections provide more details on each of these methods and the recommended best practices for password hashing. It's strongly discouraged to use MD5 or Type-7 encryption for password security.

Type-6 Encryption

For 2-way encryption/decryption requirements, IOS XR supports Type-6 encryption method. The master key used for this is protected by the TAm chip. For any key strings configured in IOS XR, customers are advised to use Type-6 encryption method.

Please refer to the configuration guide here for more details on the feature.

Password Hashing Methods

The recommended options for one-way password hashing are Type-8, 9, 10 methods.
Type-8 uses SHA-256 with PBKDF2. It uses 20,000 iterations of SHA-256.
Type-9 uses scrypt for key derivation.
Type-10 uses SHA-512.

For type-8, type-9, please refer to the configuration guide here.
For type-10 method, please refer to the configuration guide here.

Stronger Password Policies

In addition to adopting a stronger hashing mechanism, customers can now configure a password policy for each user account. Below is a snippet of the password policy configuration and the options available under password policy. For more details, please refer to the configuration guide here.

Configuring a password policy

```
RP/0/RP0/CPU0:router(config)#aaa password-policy test-policy
RP/0/RP0/CPU0:router(config-aaa)#min-length 8
RP/0/RP0/CPU0:router(config-aaa)#max-length 15
RP/0/RP0/CPU0:router(config-aaa)#lifetime months 3
RP/0/RP0/CPU0:router(config-aaa)#min-char-change 5
RP/0/RP0/CPU0:router(config-aaa)#authen-max-attempts 3
RP/0/RP0/CPU0:router(config-aaa)#lockout-time days 1
RP/0/RP0/CPU0:router(config-aaa)#commit
```

Attaching the password policy to a user account

```
RP/0/RP0/CPU0:router(config)#username user1 password-policy test-policy
password 0 pwd1
```

User Management

Administrators should control access to resources or devices. When a request for access to a resource or device is received, the request is challenged for verification of the password and identity, and access can be granted, denied, or limited based on the result. Cisco IOS XR Software provides tools to allow for multiple levels of permissions using the concepts of task and user groups. User groups are defined to have access to a certain set of capabilities. Some of these capabilities are debug commands, show commands, and configuration commands. Different user groups have configuration access to different parts of the router. A *root–system* user must be created. The *root-system* user is the most powerful user in the

Cisco IOS XR scheme and is essentially the same as a fully enabled (privilege level 15) user in Cisco IOS Software. The configuration for a *root-system* user follows:

```
username user-a
 password 7 07062F5F4B0A0C1712020A1F17253C362C
 group sysadmin
```

Cisco IOS XR Software allows the system administrator to configure groups of users and the job characteristics that are common to those groups. Groups must be explicitly assigned to users. Users are not assigned to groups by default. A user can be assigned to more than one group.
The following is a list of task groups that define different privileges for users in Cisco IOS XR Software. Each task group consists of the list of permitted tasks (*read*, *write*, *execute*, or *debug* permissions) for a user in the particular task group. Only *root-system* users have, by default, permission to run all tasks except the ones defined in the *cisco-support* group.

- *cisco-support*: Debugging and troubleshooting features, usually used by Cisco support personnel.
- *netadmin*: Configuration tasks, such as those for routing protocols.
- *operator*: Day to day monitoring activities and limited configuration rights.
- *root-system*: Configuration and display rights for all secure domain routers in the system.
- *root-lr*: Configuration and display rights for a specific secure domain router.
- *sysadmin*: Administrative tasks such as maintaining the location for stored core dumps or setting up NTP.
- *serviceadmin*: Service administration tasks.

A user group can derive attributes from another user group. Similarly, a task group can derive attributes from another task group.

Task Groups are defined by a collection of task IDs. Task groups contain task ID lists for each class of action. Each user group is associated with a set of task groups that are applicable to users in that group. A user's task permissions are derived from the task groups that are associated with the user groups to which that user belongs.

The following example shows how to create a task group named *mgmt* that is inheriting attributes from the *sysadmin* task group. *Read* permissions are assigned for any CLI commands that are associated with task ID *bgp*.

```
taskgroup mgmt
 description backbone support functions
 inherit taskgroup sysadmin
 task read bgp
```

See the Configuring AAA Services on Cisco IOS XR Software section of the Cisco IOS XR System Security Configuration Guide for more information about configuring task groups.

Administrative Access

Administrative access to the system can be lost if administrators do not fully understand or carefully plan for the following operations. A lockout of all *root-system* users is a serious issue that requires a system reload to recover the password.
Configuring authentication that uses remote AAA servers that are not available, particularly authentication for the console.
Removing the flash card from disk0:, or a disk corruption can affect certain system debugging abilities. However, if the console is available, the system is still accessible.
Configuring command authorization or EXEC authorization on the console should be done with extreme care, because TACACS+ servers may not be available or may deny every command, thus locking the user out. In particular, this lockout can occur if the authentication takes place with an unknown TACACS+ server user, or if the TACACS+ user has most or all the commands denied.
To avoid a lockout, administrators are advised to perform one of the following actions:
Before turning on TACACS+ command authorization or EXEC authorization on the console, ensure that the user who is configuring the authorization is logged in with the appropriate user permissions in the TACACS+ profile.
If the security policy of the site permits, use the **none** option for command authorization or EXEC authorization. In the event TACACS+ servers are not accessible, this options enables AAA to roll over to the *none* method, permitting the user to run the command.
In the event of failure of the TACACS+ or RADIUS services, locally configure passwords for access.

Disable Unused Services

As a security best practice, administrators should disable unnecessary services. Most services are disabled by default in Cisco IOS XR Software; however, these services can be enabled by issuing their respective configuration commands. Administrators are advised to disable the following services if they are not necessary for business operations.

TCP and UDP small services:
echo (port number 7)
discard (port number 9)
daytime (port number 13)
chargen (port number 19)
HTTP
Cisco Discovery Protocol (CDP)
Simple Network Management Protocol (SNMP)
TFTP
DHCP

To disable TCP and UDP small services, issue the following command:
no service {ipv4|ipv6} tcp-small-servers
no service {ipv4|ipv6} udp-small-servers

To disable HTTP services, issue the following command:

no http server

Cisco Discovery Protocol (CDP) can be disabled globally or under interface. To disable CDP services, issue the following command:
no cdp
interface pos 0/0/0/1
no cdp

To disable Simple Network Management Protocol (SNMP) services, issue the following command:
no snmp-server

To disable TFTP services, issue the following command:
no tftp ipv4 server
no tftp ipv6 server

To disable DHCP services if DHCP relay services are not required, issue the following command:
no dhcp {ipv4|ipv6}

If a service is required, administrators can leverage the MPP feature to enable and disable a service on the specified interface. See the Management Plane Protection section of this guide for more information.

Set Exec Timeout

By default, console, vty, and tty sessions disconnect after 10 minutes of inactivity. Administrators are advised to maintain this value at 10 minutes or less but greater than zero. A 0-minute value will prevent sessions from terminating.

Issue the following command to set an exec-timeout value:
line {console|default|template} exec-timeout (minutes) (seconds)

Issue the following command to reinstate the default timeout value:
no line {console|default|template} exec-timeout

Management Interfaces

The management plane of a Cisco IOS XR device is accessed in-band or out-of-band on a physical or logical management interface. Ideally, both in-band and out-of-band management access exists for each network device so that the management plane can be accessed during network outages.
One of the most common interfaces used for in-band access is the logical loopback interface. Loopback interfaces are always available, whereas physical interfaces can change state, resulting in the interface and subsequently the device not being accessible. Administrators are advised to add a loopback interface to each device as a management interface and to use it exclusively for the management plane.

In addition to loopback interfaces, administrators are advised to use the Virtual Address feature in combination with uniquely addressed management interfaces:

```
ipv4 virtual address 12.7.49.171 255.255.0.0
 "Virtual (shared) Address"
 !
 interface MgmtEth0/RP0/CPU0/0       "Primary RP"
  ipv4 address 12.7.49.172 255.255.0.0
 !
 interface MgmtEth0/RP1/CPU0/0       "Secondary RP"
  ipv4 address 12.7.49.173 255.255.0.0
 !
```

To ensure that UDP traffic flow, such as syslog and SNMP traps, is not interrupted during the failover, administrators are advised to use virtual addresses to source all management traffic, as shown in the following configuration:

```
ipv4 virtual address use-as-src-addr
```

IPsec for Management Plane Encryption

Starting with IOS XR 7.8.1 release, IPsec can be enabled to protect entire management plane traffic. More details on the feature and configuration guide can be found here.

Platform Specific Note
This feature is supported only on N540X-12Z16G-SYS-A variant of NCS 540 series routers.

Limit Network Access with Access Control Lists

Devised to prevent unauthorized direct communication to network devices, infrastructure ACLs are one of the most critical security control mechanisms that can be implemented in the network.
An ACL is constructed and applied to specify necessary connections between hosts or networks and network devices. Common examples of these types of connections are eBGP, SSH, and SNMP. After the required connections have been permitted, all other traffic to the infrastructure is explicitly denied. All transit traffic that crosses the network and is not destined to infrastructure devices is explicitly permitted.
The protection provided by ACLs is relevant to both the management and control planes. The implementation of ACLs can be made easier using distinct addressing for network infrastructure devices.
The following ACL configuration example illustrates the structure that is required as a basis for starting the ACL implementation process:

```
ipv4 access-list ACL-INFRASTRUCTURE-IN
 !—-Permit required connections for routing protocols and
 10 permit tcp host <trusted-ebgp-peer> host <local-ebgp-address> eq 179
```

```
20 permit tcp host <trusted-ebgp-peer> eq 179 host <local-ebgp-address>
30 permit tcp host <trusted-management-stations> any eq 22
40 permit udp host <trusted-netmgmt-servers> any eq 161
!—Deny all other IP traffic to any network device
!
50 Deny ipv4 any <infrastructure-address-space> <mask>
!—Permit transit traffic
!
60 permit ipv4 any any
```

When created, the ACL must be applied to all interfaces that face non-infrastructure devices, including interfaces that connect to other organizations, remote access segments, user segments, and segments in data centers.

Securing Interactive Management Sessions

Management sessions allow administrators to view and collect information about a Cisco IOS XR device and its operations. If this information is disclosed to a malicious user, the device can become the target of an attack or used as a source of additional attacks. Anyone with privileged access to a Cisco IOS XR device has the capability for full administrative control of the device. It is imperative to secure management sessions to prevent information disclosure and unauthorized access.

Management Plane Protection

The Management Plane Protection (MPP) feature in Cisco IOS XR Software provides the capability to restrict the interfaces on which network management packets are allowed to enter a device. MPP allows a network operator to reserve a set of interfaces for management traffic either exclusively, or along with regular data. MPP changes the default behavior of an interface by not listening to all the services traffic and allowing traffic only from applications that are explicitly configured on the interface.
MPP provides the support for in-band management interfaces. An in-band interface shares management and forwarding traffic.
MPP provides a mechanism for specifying configurable out-of-band interfaces. An out-of-band management interface receives network management traffic. The advantage of this implementation is that network management traffic does not interfere with any forwarding or customer traffic, significantly reducing the possibility of potential side effect, such as garbled packet and a denial of service (DoS) condition, of processing network management traffic. MPP also supports peer-filtering mechanisms. Management traffic that belongs to the configured peer ip-addresses would be allowed on a specified interface; other traffic would be dropped.
In-band management interface: This Cisco IOS XR physical or logical interface processes management packets, as well as data-forwarding packets. An in-band management interface is also called a *shared management interface*.
Out-of-band management interface: This interface allows only management protocol traffic to be forwarded or processed. An *out-of-band management interface* is defined by the network operator to specifically receive network management traffic. The advantage is that

forwarding (or customer) traffic cannot interfere with the management of the router, which significantly reduces the possibility of DoS attacks.
Out-of-band interfaces: This interface forwards traffic only between out-of-band interfaces or terminate management packets that are destined to the router. In addition, the out-of-band interfaces can participate in dynamic routing protocols. The service provider connects to the router's out-of-band interfaces and builds an independent overlay management network, with all the routing and policy tools that the router can provide. Management ports are out-of-band by default.
Device management traffic may enter a device only through these management interfaces. After MPP is enabled, no interfaces except the designated management interfaces will accept network management traffic that is destined to the device. Restricting management packets to the designated interfaces provides greater control over the management of a device, providing more security for that device. Administrators are advised to use the **peer-filtering** option to control management traffic from specific peers or a range of peers.
MPP configuration does not enable the specific management protocol services. MPP is responsible only for making the services available on different interfaces. Currently, MPP only controls the incoming management requests for protocols, such as TFTP, Telnet, Simple Network Management Protocol (SNMP), Secure Shell (SSH), and HTTP and HTTPS.

The following example shows how to enable the MPP to only allow SSH and HTTPS requests from peer address 10.0.1.0 on the GigabitEthernet0/0/01 interface:

```
control-plane
 management-plane
 inband
 interface Gig 0/0/01
 allow ssh peer address ipv4 10.0.1.0/24
 allow https peer address ipv4 10.0.1.0/24
```

The following example shows how to configure GigabitEthernet0/2/0/0 as an out-of-band interface and only allow a SSH request from peer address 10.10.10.10:

```
control-plane
  management-plane
  out-of-band
  interface GigabitEthernet0/2/0/0
  allow SSH peer
  address ipv4 10.10.10.10
```

Note: In Cisco IOS XR version 3.8.0 and later, MPP is enabled on all interfaces by default. Any TCP and UDP port can be accessed from any interface. However, when MPP is configured, access is only permitted on management interfaces until the MPP permit is applied to other interfaces.
For more information on management plane protection, see the Implementing Management Plane Protection on Cisco IOS XR Software section in the Cisco IOS XR System Security Configuration Guide.

Control and Encrypt Management Sessions

Because information can be disclosed during an interactive management session, traffic must be encrypted so that a malicious user cannot access the data that is being transmitted. Encrypting the traffic allows a secure remote access connection to the device. If the traffic for a management session is sent over the network in plain text, an attacker could obtain sensitive information about the device and the network.

Telnet Protocol

Although popular, Telnet is not a secure protocol, and administrators of Cisco IOS XR devices are advised not to use Telnet. The use of Telnet on Cisco IOS XR devices will require special attention.

The Cisco Internet services process daemon, *Cinetd*, which is similar to the UNIX daemon, *inetd*, is a multi-threaded server process that is started by the system manager after the system has booted. *Cinetd* listens on a well-known port on behalf of the server program. When a service request is received on the particular port, *Cinetd* notifies the server program that is associated with the service request. By default, *Cinetd* is not configured to listen for any services.

Telnet service is disabled by default on Cisco IOS XR. The following command is required to enable Telnet service on a Cisco IOS XR device:

telnet [ipv4 | ipv6 | vrf vrf-name] server max-servers [option]

<1-200> Set number of allowable Telnet servers

no-limit No limit to number of allowable Telnet servers

Two options are available to define the maximum allowable Telnet servers and to set "no limit" on the number of allowable Telnet servers.

Note: Administrators are advised not to choose the **no-limit** option, because the option could bring the system to an unstable state. Cisco IOS XR Software versions 3.8 and later do not include the **no-limit** option.

To add protection using Telnet service, the following configuration is recommended:

Restrict Telnet access using ACL, MPP, and peer control.

Estimate the maximum Telnet sessions that the system may need to set Telnet max-servers accordingly.

Add a hardware rate-limit for Telnet packets using the Local Packet Transport Services (LPTS) feature, if necessary. The default value of TELNET-known is 1000 packets per second (p/s). The default value of TELNET-default is 500 p/s. The default value can be changed to a lower value as shown in the following example:

```
# below is global configure across all LCs
lpts pifib hardware police
  flow telnet default rate 200
  flow telnet known rate 200
# below is specific LC configure which has precedence than global configure
lpts pifib hardware police location 0/1/CPU0
  flow telnet default rate 50
  flow telnet known rate 200
```

Note: The 200 p/s is a sample number for reference only.

This feature is available in Cisco IOS XR Software versions 3.6 and later.

For more information about securing Telnet protocols, see the Cisco IOS XR IP Addresses and Services Configuration Guide and Cisco IOS XR IP Addresses and Services Command Reference.

Platform Specific Note
Telnet package is now explicitly taken out from the base images of the below mentioned IOS XR based platforms.

- All variants of Cisco 8000
- N540X-16Z4G8Q2C-D/A
- N540X-16Z8Q2C-D
- N540-28Z4C-SYS-D/A
- N540X-12Z16G-SYS-D/A
- N540-12Z20G-SYS-D/A
- N540X-4Z14G2Q-D/A
- N540X-8Z16G-SYS-D/A
- N540X-6Z18G-SYS-D/A
- N540-6Z18G-SYS-D/A
- N540-6Z14S-SYS-D/A
- N540-FH-AGG-SYS
- N540-FH-CSR-SYS

SSH Protocol

Users can establish an encrypted and secure remote access management connection to a device by using the SSH, SFTP, or HTTPS protocols. Cisco IOS XR Software supports SSH Version 1.0 (SSHv1), SSH Version 2.0 (SSHv2), and HTTPS that uses SSL and Transport Layer Security (TLS) for authentication and data encryption. It is recommended to only enable SSH version 2.0.

The following example configuration enables SSHv2 on a Cisco IOS XR device:
hostname test-1
domain name test.com
ssh server v2

Cisco IOS XR Software also supports SFTP, which provides an encrypted, secure, and authenticated connection for copying device configurations or software images. SFTP relies on SSHv2. After SSHv2 has been configured, the SFTP feature is available on the router.
Both versions of SSH require more RP CPU resources than Telnet that could cause higher RP CPU usage during high-rate SSH requests or an SSH DoS attack. The following configuration can help prevent excessive RP CPU usage:

Restrict the SSH access using MPP and peer control

In Cisco IOS XR, the default rate-limit of the SSH server is 60 requests per minute; users can change this rate to a lower value.

ssh server rate-limit 10

Rate-limit the SSH packets in the hardware layer using LPTS. The default value is 1000 p/s for Cisco IOS XR Software versions 3.6 and later. This value can be changed to a lower value, as shown in the following example:

```
# below is global configure across all LCs
lpts pifib hardware police
  flow ssh default rate 200
  flow ssh known rate 200

# below is is specific LC configure which has precedence than global configure
lpts pifib hardware police location 0/1/CPU0
  flow ssh default rate 50
  flow ssh known rate 200
```

Passwordless SSH

To further harden the security posture of the devices using SSH, it's recommended to move out of password-based authentication and adopt passwordless options for SSH connections. Cisco's IOS XR releases beyond 7.x support public-key based SSH authentication. In addition to this, X.509v3 certificate based SSH authentication is supported from 7.3.1 release.

For RSA public key-based authentication, IOS XR supports importing the public key of the users. More details can be found in the configuration guide here.

For customers who don't prefer managing the keys for each of the users, certificate-based authentication can be used. The only restriction of the feature is that only local authentication is supported but authorization can be through a remote TACACS server. More details on the certificate based SSH authentication can be found in the configuration guide here.

HTTP and HTTPS

Cisco IOS XR Software may use the craft web interface (CWI) to support remote configuration and monitoring. In general, HTTP access is equivalent to interactive access to the router. The authentication protocol used for HTTP is equivalent to sending a plaintext password across the network and there is no effective provision in HTTP for challenge-based or one-time passwords. HTTP access should be restricted to appropriate IP addresses that use the **http server access-group** command or MPP with peer control. As with interactive logins, the best choice for HTTP authentication is a TACACS+ or RADIUS server. Administrators are advised to avoid using "enable" as the password for the HTTP authentication.
HTTPS (HTTP over SSL or HTTP Secure) is the use of SSL or Transport Layer Security (TLS) as a sub-layer under regular HTTP application layering. HTTPS encrypts and decrypts user page requests as well as the pages that are returned by the web server. Because HTTPS protects against eavesdropping and man-in-the-middle attacks, administrators are advised to use HTTPS instead of HTTP.

HTTPS can be enabled only after SSL initialization. To initialize SSL, Rivest, Shamir, and Adelman (RSA) or Digital Signature Algorithm (DSA), the following actions need to be performed:

1. The key pairs need to be generated.
2. A Certificate Authority (CA) needs to be enrolled.
3. A CA certificate for the router key needs to be obtained.

The following example shows how to generate the RSA keys for the router, configure a trust point, authenticate to the CA server, and obtain a key certificate from the CA:

```
crypto key generate rsa general-keys
configure
domain ipv4 host xyz-ultra5 10.0.0.5
crypto ca trustpoint myca
enrollment url http://xyz-ultra
!
crypto ca authenticate myca
crypto ca enroll myca
```

The following example shows how to enable HTTPS:

```
http server ssl
```

Control Physical and Virtual Terminals

In Cisco IOS XR devices, physical and virtual terminals are "lines" that can be used for local and remote access to a device.

Note: The console ports on Cisco IOS XR devices have special privileges that allow an administrator to perform the password recovery procedure. An unauthenticated attacker who is trying to perform password recovery will require access to the console port and be able to interrupt power to the device or cause a crash of the device.

Any method that is used to access the console port of a device must be secured in a manner that is equal to the security that is enforced for privileged access to a device. Methods used to secure access must include the use of AAA, exec-timeout, and modem passwords (if a modem is attached to the console).

The Cisco IOS XR Software assigns a vty identifier to the vty lines and connections according to the order in which the vty connection was established. A vty line is used for remote network connections that are supported by the device regardless of protocol (for example, SSH, SCP, or Telnet). To ensure that a device can be accessed by way of a local or remote management session, proper controls must be enforced on the vty lines. Cisco IOS XR devices have a limited number of vty lines; the number of lines available can be determined by issuing the **show line** EXEC command. When all the vty lines are in use, new management sessions cannot be established, resulting in a DoS condition on the device.

The terminals are configured by using line templates. The following line templates are available in the Cisco IOS XR Software:

Default line template: This template applies to all physical and virtual terminal lines.

Console line template: This template applies to the console line.

Line templates: This is a user-defined line template that can be applied to a range of virtual terminal lines.
Attributes that are not defined in the console template or in any virtual template are taken from the default template. Administrators can secure the console by using the following configuration:
line console
exec-timeot 60 0
secret s3cr3t

The simplest form of access control to the vty of a device is using authentication on all lines, regardless of the device location within the network. This authentication is critical for vty lines because they are accessible by way of the network. Other forms of vty access controls can be enforced by using the **transport input** or **access-class** configuration commands, using the MPP and LPTS features, or by applying ACLs to the physical interfaces on the device. Authentication can be enforced using AAA through the local user database or by simple password authentication configured directly on the vty line. AAA is the recommended method for authenticated access to a device.

The **exec-timeout** command must be used to log out sessions on vty lines that are left idle. A vty should be configured to accept only encrypted and secure remote access management connections to the device using the following configuration:

line {default | console | temple}
transport input ssh

vty lines allow an administrator to connect to other devices. Administrators are advised to use the transport output line configuration command to limit the type of transport outgoing connections. If outgoing connections are not necessary, then *transport output none* should be used. However, if outgoing connections are allowed, an encrypted and secure remote access method for the connection should be enforced using transport output SSH.

The following is an example that shows a configuration for securing vty lines:
line default
 access-class 20 in
 exec-timeout 60 0
 password s3cr3t
 transport preferred ssh

Any vty should be configured to accept connections with only the protocols that are necessary. Configuration can be done using the **transport input** command. For example, a vty that is expected to receive only Telnet sessions would be configured with transport input Telnet, while a vty that permits both Telnet and SSH sessions would have transport input Telnet SSH. If the software supports an encrypted access protocol, such as SSH, administrators are advised to enable only that protocol and to disable plaintext Telnet. Administrators also may consider using the **ip access-class** command to restrict the IP addresses from which the vty will accept connections. Again, MPP and LPTS are the

recommended solutions for limiting packet rate to the system at a hardware level. ACLs should be used as a secondary protection in case MPP and LPTS are not configured properly.

Control vty's and Ensure vty Availability

Cisco IOS XR supports up to 100 vty lines (its default number of vty lines is five), and they are managed by configuring vty-pool template (vty-pool default 0 99). When all the vtys are in use, further remote interactive connections cannot be established, creating the opportunity for a DoS attack; if an attacker can open remote sessions to all the system vty's, the legitimate administrator may not be able to log in. An exploit does not require the attacker to log in using a valid username and password because the sessions can be left at the login prompt.

Configuring a more restrictive **ip access-class** command on the last system vty may reduce the likelihood of an attack. The last vty could be restricted to accept connections from a single, specific administrative workstation, whereas the other vtys may accept connections from any address in a corporate network.

The following configuration example shows that vty ports 5 and 6 accept connections only from IP source address 10.10.10.10 (for example, administrator host), while ports 0–4 can accept connections from any source.

```
ipv4 access-list vty5-permit
 10 permit ipv4 host 10.10.10.10 any
 20 deny ipv4 any any
 !

line template vty5
 password 7 0822455D0A16544541
 access-class ingress vty5-permit
!

vty-pool vty5 5 6 line-template vty5

vty-pool default 0 4
```

Cisco IOS XR devices have default vty timeouts of 30 seconds if there is no response on a login prompt. These timeouts prevent idle sessions from consuming a vty indefinitely and also have default timeout if any open vty sessions are idle for more than 5 minutes. Because there is no TCP keep alive command in Cisco IOS XR Software, these timeouts help guard against both malicious attacks and "orphaned" sessions that are caused by remote system crashes.

Complete vty protection can be provided by disabling all non-IP based remote access protocols and using IPsec encryption for all remote interactive connections to the router.

Warning Banners

In some jurisdictions, it may be impossible to prosecute malicious users and illegal to monitor them unless they have been notified that they are not permitted to use the system. One method of notification is to place this information into a banner message that is configured using the Cisco IOS XR Software banner login command.

Legal notification requirements are complex, vary by jurisdiction and situation, and should be discussed with legal counsel. Legal opinions can vary within the same jurisdiction. In cooperation with legal counsel, a banner should provide some or all the following information:

1. Notice that the system is to be logged into or used only by specifically authorized personnel with information about who can authorize to use.
2. Notice that unauthorized use of the system is unlawful and may be subject to civil and criminal penalties.
3. Notice that use of the system can be logged or monitored without further notice and that the resulting logs can be used as evidence in court.
4. Specific notices required by local laws.

From a security point of view, a login banner should not contain any specific information about the router name, model, software, or ownership. This information can be exploited by malicious users. The following is an example of the banner configuration where "c" is a delimiting character:
banner motd c This is restricted to access c

Authentication, Authorization, and Accounting

The Authentication, Authorization, and Accounting (AAA) framework is critical to securing interactive access to network devices. The AAA framework provides a highly configurable environment that can be tailored depending on the needs of the network.
Authentication is the most important security process by which a principal (a user or an application) obtains access to the system. The principal is identified by a username or user ID that is unique across an administrative domain. The applications serving the user, such as EXEC or Management Agent, obtain the username and the credentials from the user. AAA performs the authentication based on the username and credentials that are passed to it by the applications. The role of an authenticated user is determined by the group (or groups) to which the user belongs. A user can be a member of one or more user groups.

A user is a *root-system* user if they belong to the *root-system* group. The *root-system* user may be defined in the local or remote AAA database.

AAA can be enabled on a Cisco IOS XR device using a configuration like the following example:
aaa authentication login default group tacacs+ local

The following example shows how to enable the Remote method using the admin configure mode:
aaa authentication login remote group tacacs+ local
See the Configuring AAA Services on Cisco IOS XR Software section of the Cisco IOS XR System Security Configuration Guide for more information about AAA.

TACACS+ and RADIUS Authentication

Cisco IOS XR Software communicates with the remote AAA server using a standard IP-based security protocol, such as TACACS+ or RADIUS.

TACACS+ uses a TCP port (49), while RADIUS uses UDP. TACACS+ provides reliable data transfer between the client and server; RADIUS provides faster operation. Benefits to using TACACS+ include the following:

1. TACACS+ allows true command authorization. Users can create clear usage policies with TACACS+, whereas different users have access to different commands with administrative granularity. TACACS+ can do this because it separates the Authentication and Authorization functions. RADIUS combines these functions.
2. TACACS+ encrypts the entire payload of the client-server exchange. This action is important for the protection of highly secure environments. In contrast, RADIUS encrypts only the password, allowing intercepted packets to reveal important information.

The strongest point in favor of RADIUS is that it is an open standard that is implemented by many vendors, including Cisco.
For more information about the differences between RADIUS and TACACS+, see TACACS+ and RADIUS Comparison.

The following sample configuration shows how TACACS+ authentication can be enabled on a Cisco IOS XR device:

```
tacacs-server host 10.1.1.1 port 49
  key cisco123
tacacs source-interface Loopback0
```

The following sample configuration shows how RADIUS authentication can be enabled on a Cisco IOS XR device:

```
radius-server host 5.5.151.1 auth-port 1812 acct-port 1813
 key cisco123
 !
```

Authentication Fallback

If all configured TACACS+ servers become unavailable, a Cisco IOS XR device can rely on secondary authentication protocols. Typical configurations include the use of local database authentication if all configured TACACS+ servers are unavailable.
The complete list of options for on-device authentication includes local or line. Line authentication uses passwords that are configured under line device statements such as console or default templates, while local authentication sets passwords that can be used on all lines. Both options have advantages.

There are two options for password configuration: the use of clear text via the password command, and the use of encrypted secure password by configuring the secret option. The secret option is preferred. More details on password protection can be found in this section.

If TACACS+ became completely unavailable, each administrator could use their local username and password. Although this action enhances the accountability of network administrators during TACACS+ outages, it significantly increases the administrative burden, because local user accounts on all network devices must be configured and subsequently maintained.

The following configuration example builds on the previous TACACS+ authentication example to include fallback authentication to the locally configured password, or to the line password if no local password is configured:
aaa authentication login default group tacacs+ local line

Redundant AAA Servers

Whether TACACS+ or RADIUS servers are used, the AAA servers should be redundant and deployed in a fault-tolerant manner. These attributes can help ensure that interactive management access such as SSH is possible if an AAA server is not available. Administrators are advised to consider the following factors when designing AAA servers:

1. Availability of AAA servers during potential network failures
2. Geographically dispersed placement of AAA servers
3. Load on individual AAA servers during steady-state and failure conditions
4. Network latency between Network Access Servers and AAA servers
5. AAA server databases synchronization

The following example shows how to create redundancy by configuring a total of three TACACS+ servers to be used by the Cisco IOS XR device:
tacacs-server host 1.1.1.1 port 1 key abc
tacacs-server host 2.2.2.2 port 2 key def
tacacs-server host 3.3.3.3 port 3 key ghi
aaa group server tacacs+ tacgrp
server 1.1.1.1
server 2.2.2.2
server 3.3.3.3

Fortify Simple Network Management Protocol

This section highlights methods that can be used to secure the deployment of Simple Network Management Protocol (SNMP) within Cisco IOS XR devices. SNMP must be properly secured to protect the confidentiality, integrity, and availability of the network data and the network devices through which the data transits. SNMP provides a wealth of information on the health of network devices. This information should be protected from malicious users who may want to take advantage of the data for use in attacks against the network.

SNMP Community Strings

Community strings are passwords that are applied to a Cisco IOS XR device to restrict access, both *read-only* and *read-write*, to the SNMP data on the device. Community strings, as with all passwords, should be carefully chosen to conform to security best practices. Community strings should be changed at regular intervals and in accordance with network security policies. For example, the strings should be changed when a network administrator changes roles or leaves the company.
The following example shows a sample configuration for a global SNMP community string, *s3cr3t*:
snmp-server community s3cr3t

The following configuration lines illustrate the configuration of a *read-only* community string of *r3adm3* and a *read-write* community string of *s3cr3t*:

```
snmp-server community r3adm3 RO
snmp-server community s3cr3t RW
```

Note: The preceding community string examples have been chosen to clearly explain the use of community strings. For production environments, community strings should be chosen with caution and should consist of a series of alphabetical, numerical, and non-alphanumeric symbols.

To limit SDR SNMP access, the SDRowner or SystemOwner options can be used in the following manner:

```
snmp-server community r3adm3 RO SDRowner
snmp-server community r3adm3 RO SystemOwner
```

SNMPv2 Community strings are displayed in plain text. Administrators are advised to change community strings on a periodic basis as changing passwords.

SNMP Community Strings with Access Control Lists

In addition to the community string, an ACL should be applied that further restricts SNMP access to a select group of source IP addresses. The following configuration example restricts SNMP *read-only* access to end host devices that reside in the 192.168.100.0/24 address space and restricts SNMP *read-write* access to only the end host device at 192.168.100.1.

Note: The devices that are permitted by these ACLs still require the proper community string to access the requested SNMP information.

```
ipv4 access-list SNMP-ROACCESS
10 permit ipv4 192.168.100.0 0.0.0.255 any
ipv4 access-list SNMP-RWACCESS
10 permit ipv4 host 192.168.100.1 any
snmp-server community r3adm3 RO SNMP-ROACCESS
snmp-server community s3cr3t RW SNMP-RWACCESS
```

Control SNMP Access with ACLs

Infrastructure ACLs can be deployed to ensure that only end hosts with trusted IP addresses can send SNMP traffic to a Cisco IOS XR device. An ACL should contain a policy that denies unauthorized SNMP packets on UDP port 161.

Control SNMP with Management Plane Protection

The Management Plane Protection (MPP) feature in Cisco IOS XR Software can be used to help secure SNMP by restricting the interfaces through which SNMP traffic can terminate on the device. The following example shows hundredGigE 0/0/0/24 as the interface that will terminate SNMP traffic from host 10.1.1.1:

```
control-plane
 management-plane
 inband
 interface hundredGigE 0/0/0/24
```

```
allow snmp
allow snmp peer
 address ipv4 10.1.1.1
```

SNMP Views

SNMP Views is a security feature that can permit or deny access to certain SNMP MIBs. When a view is created and applied to a community string with the *snmp-server community community-string view* global configuration commands, access to MIB data is restricted by the permissions that are defined with the view. When appropriate, users are advised to use views to limit access to specific data to SNMP users.
The following configuration example restricts SNMP access with the community string *LIMITED* to the MIB data that is located in the VIEW−SYSTEM−ONLY group, as configured in the first three lines of the following example using the **snmp-server view** command:

```
snmp-server view VIEW-SYSTEM-ONLY ciscoPingMIB included
snmp-server view VIEW-SYSTEM-ONLY sysUpTime included
snmp-server view VIEW-SYSTEM-ONLY system include
snmp-server community LIMITED view VIEW-SYSTEM-ONLY RO
```

SNMP Version 3

SNMP Version 3 (SNMPv3) is defined by RFCs 3410 through 3415 and is an interoperable, standards-based protocol for network management. SNMPv3 provides secure access to devices by authenticating and optionally encrypting packets over the network. Where supported, SNMPv3 can be used to add another layer of security when deploying SNMP.
SNMPv3 consists of three primary configuration options:
no auth: This mode does not require authentication or encryption of SNMP packets.
auth: This mode requires authentication of the SNMP packets without encryption.
priv: This mode requires both authentication and encryption (privacy) of each SNMP packet.
An authoritative engine ID must exist to use the SNMPv3 security mechanisms (authentication or authentication and encryption) for handling SNMP packets; by default, the engine ID is generated locally. The engine ID can be displayed with the show **snmp engineID** command, as shown in the following example:

```
#show snmp engineID
SNMP engineID 00000009000000a1ffffffff
```

Note: If the engineID is changed, all SNMP user accounts must be reconfigured.
The next step is to configure a SNMPv3 group. The following example shows the configuration of a Cisco IOS XR device for SNMPv3 with an SNMP server group *AUTHGROUP* and enables authentication only for this group by using the **auth** keyword:

```
snmp-server group AUTHGROUP v3 auth
```

The following example shows the configuration of a Cisco IOS device for SNMPv3 with an SNMP server group *PRIVGROUP* and enables both authentication and encryption for this group by using the **priv** keyword:

```
snmp-server group PRIVGROUP v3 priv
```

The following example shows the configuration of an SNMPv3 user *snmpv3user* as part of the group *snmpusers* with a Message-Digest algorithm 5 (MD5) authentication password of*authpassword* and a 3DES encryption password of *privpassword*:
snmp-server user snmpv3user snmpusers v3 auth md5 authpassword priv 3des privpassword

Protect SNMP Private Community Strings

SNMP private community strings should use complex expressions to prevent guessing or hacking by attackers. For Cisco IOS XR Software Release versions prior to 3.8, the following SNMP configuration must be added to prevent SNMP private community strings from being retrieved:
snmp-server view novacm internet included
snmp-server view novacm internet.6.3.16 excluded
snmp-server community public view novacm RO

Logging Best Practices

Event logging provides users visibility into the operation of a Cisco IOS XR device and the network on which the device is deployed. Cisco IOS XR Software provides several flexible logging options that can help achieve the network management and visibility goals of an organization.
The following sections provide some basic logging best practices to help administrators utilize logging successfully while minimizing the impact of logging on a Cisco IOS XR device. See the Implementing Logging Services section of the Cisco IOS XR System Monitoring Configuration Guide for more information about logging configuration.

AAA Logging

AAA logging collects information about user dial-in connections, logins, logouts, HTTP access, privilege-level changes, commands executed, and similar events. AAA log entries are sent to authentication servers using the TACACS+ and RADIUS protocols and are recorded locally by those servers, typically in disk files. If using a TACACS+ or RADIUS server, administrators are advised to enable AAA logging of various types; this is done using AAA configuration commands such as AAA accounting. For details, see the Authentication, Authorization, and Accounting section of this guide.

Access Control List Violation Logging

When using access lists to filter traffic, administrators are advised to log some packets that violate the filtering criteria to find out what type of traffic is being sent to the router. Because access list log messages are rate-limited, the performance impact on Cisco IOS XR devices is minimal.
The following ACL logging example is used to log access violation from class A address 10.0.0.0 to host 202.202.202.20 and includes input interface:
ipv4 access-list violation-log
 10 deny ipv4 10.0.0.0 0.255.255.255 host 202.202.202.20 log-input

20 permit ipv4 any any
!

Logging Correlation

Logging correlation can be used to isolate the most significant root messages for events affecting system performance.
See the Implementing and Monitoring Alarms and Alarm Log Correlation section of the Cisco IOS XR System Monitoring Configuration Guide for more information about logging.

Send Logs to a Central Location

Administrators are advised to send logging information to a remote syslog server to more effectively correlate and audit network and security events across network devices.
Note: Syslog messages are transmitted unreliably by UDP and are transmitted in plain text. For Any protections that a network affords to management traffic, for example, encryption or out-of-band access, should be extended to include syslog traffic.
The following example shows the configuration of Cisco IOS XR device sending logging information to a remote syslog server with the IP address 10.1.1.1 with default udp port 514:
logging 10.1.1.1 port default
!

Logging Levels

Each log message generated by a Cisco IOS XR device is assigned one of eight severity levels that range from emergencies to debugging. Unless specifically required, administrators are advised to avoid logging at the debugging level. Logging at the debugging level produces an elevated CPU load that can lead to device and network instability.
The global configuration command **logging trap level** is used to specify the logging messages that are sent to remote syslog servers. The level specified indicates the lowest severity message that is sent. For buffered logging, the **logging buffered** level command is used.
The following configuration example shows how to send log messages
(from *informational* through *emergencies**)*** to the local log buffer.
logging buffered informational

Disable Console or Monitor Sessions

Cisco IOS XR Software allows administrators to send log messages to monitor sessions. Monitor sessions are interactive management sessions in which the EXEC command, **terminal monitor**, is issued to the console. The use of the **terminal monitor** command is not recommended because it can increase the CPU load of a Cisco IOS XR device. Instead, administrators are advised to send logging information to the local log buffer that can be viewed by issuing the **show logging** command.
Administrators can use the global configuration commands **logging console disable** and **logging monitor disable** to disable logging to the console and monitor sessions, as shown in the following configuration:
logging console disable

logging monitor disable

Buffered Logging

Cisco IOS XR Software supports the use of a local log buffer so that an administrator can view locally generated log messages. The use of buffered logging is recommended over the use of logging to either the console or monitor sessions.
There are two configuration options that are relevant when configuring buffered logging: the logging buffer size and the message severities that are stored in the buffer. The size of the logging buffer is configured with the global configuration command logging buffered size. The lowest severity included in the buffer is configured using the logging buffered severity command. An administrator can view the contents of the logging buffer through the **show logging** EXEC command.
The following example configures a logging buffer size of 4096000 bytes, as well as a logging severity of *informational*, indicating that messages at
levels *emergencies* through *informational* will be stored:
logging buffered 4096000
logging buffered informational
Caution: Administrators are advised to use caution when increasing the logging buffer size. The buffer size should not be set to large number in comparison to the total memory available on the device.

Configure Logging Source Interface

To provide an increased level of consistency and reliability when collecting and reviewing log messages, administrators are advised to statically configure a source interface from which all syslog messages will be sent. Accomplished with the **logging source interface** command, statically configuring the **logging source interface** ensures that the same IP address appears in all logging messages that are sent from an individual Cisco IOS XR device. For added stability, it is advisable to use a loopback interface as the logging source.
The following configuration example illustrates the use of the **logging source-interface interface** global configuration command to specify that the IP address of the loopback 0 interface be used for all outgoing syslog messages:
logging source-interface loopback 0

Configure Logging Timestamps

The configuration of logging timestamps helps correlate events across network devices. It is important to implement a correct and consistent logging timestamp configuration so that logging data can be correlated. Logging timestamps should be configured to include the date and time to millisecond precision and to include the time zone in use on the device. Syslog servers and routers also need to be synchronized using the same Network Time Protocol (NTP) clock source.
The following example shows the configuration of logging timestamps with millisecond precision within the Coordinated Universal Time (UTC) zone:
service timestamps log datetime msec show-timezone

Alternatively, it is possible to configure a specific local time zone (instead of using UTC) and configure that information to be present in generated log messages. The following example shows the configuration of a device to use the Pacific Standard Time (PST) zone in the log message timestamp:
clock timezone PST -8
service timestamps log datetime msec localtime show-timezone

Configuration Management

Cisco IOS XR Software includes several configuration management features that can be enabled. These features include functionality to back up software and configurations, the ability to roll back the configuration to a previous version, and the ability to create a detailed configuration change log.

Configuration Encryption

For data-at-protection of sensitive data like running configuration on the router, IOS XR supports a partial disk encryption feature from 7.3.1 release. Customers are encouraged to enable disk encryption feature. More details on the feature can be found in the configuration guide here.

The encryption key is protected by the TAm chip* to make sure an attacker cannot extract it to decrypt the disk even if they have physical possession of the device. IOS XR also supports a zeroization CLI+ to clear the encryption key. By clearing the encryption key from the router, even Cisco cannot decrypt the disk contents and sensitive data would be protected even on RMA or decommissioned devices.

***Platform Specific Note**
TAm chip protects the encryption key only on the below variants of IOS XR.
- All variants of Cisco 8000
- N540X-16Z4G8Q2C-D/A
- N540X-16Z8Q2C-D
- N540-28Z4C-SYS-D/A
- N540X-12Z16G-SYS-D/A
- N540-12Z20G-SYS-D/A
- N540X-4Z14G2Q-D/A
- N540X-8Z16G-SYS-D/A
- N540X-6Z18G-SYS-D/A
- N540-6Z18G-SYS-D/A
- N540-6Z14S-SYS-D/A
- N540-FH-AGG-SYS
- N540-FH-CSR-SYS

+Platform Specific Note
Currently supported only on the above mentioned IOS XR platforms.

Create Software and Configurations Backups

Currently, there are two methods to back up software and configurations for Cisco IOS XR devices: the use of disk backup, also known as "Golden Disk," and the use of the disk mirroring function.
See the Configuring Disk Backups and Disk Mirroring in Cisco IOS XR Software section of the Cisco IOS XR System Management Configuration Guide for more information about software and configuration backup.

Disk Backup

A system backup disk is created when the system files are backed up to a local storage device for the first time. This process formats the selected device and copies the software packages and system configurations to the local storage device, making it possible to boot a system or perform recovery configuration if the primary disk becomes corrupted. The following example shows how to use the disk backup utility to make a copy of the files from disk0: to disk1:
system backup disk0: disk1:

Disk Mirroring

Disk mirroring replicates the critical data on the primary boot device onto another storage device on the same RP, henceforth referred to as the secondary device. If the primary boot device fails, applications continue to be serviced transparently by the secondary device, thereby avoiding a switchover to the standby RP. The failed primary storage device can be replaced or repaired without disruption of service.
Disk mirroring should mirror only critical data on the primary boot device to a secondary storage device. Disk mirroring should not include non-critical data such as logging data. To separate critical data from non-critical data, the disk devices need to be partitioned according to the following list:
Disk0: is partitioned to disk0: and disk0a:
Disk1: is partitioned to disk1: and disk1a:
Disk0: and disk1: are used for critical data, while disk0a: and disk1a: are used for logging data and other non-critical data. Before disk mirroring configuration on the RP, the secondary storage device must be partitioned. The following example shows a disk mirroring configuration:

mirror location 0/rp0/cpu0 disk0:disk1:

Exclusive Configuration Change Access

The Cisco IOS XR Exclusive Configuration Change Access feature ensures that only one administrator can make configuration changes to a Cisco IOS XR device at a given time. This feature helps eliminate simultaneous and potentially conflicting changes being made to the device. The following example shows the exclusive configuration:

configure exclusive

Control Plane

The control plane contains the logical group of all routing, signaling, link-state, and other control protocols that are used to create and maintain the state of the network. The control plane includes the following protocols:
Open Shortest Path First (OSPF)
Intermediate System – Intermediate System (IS-IS)
The Border Gateway Protocol (BGP)
Internet Control Message Protocol (ICMP)
Label Distribution Protocol (LDP)
Resource Reservation Protocol (RSVP)
Protocol Independent Multicast (PIM)
The control plane also includes functions as the Address Resolution Protocol (ARP), the Bidirectional Forwarding Detection (BFD), and the Layer 2 keepalives that maintain interface state.
It is important that management plane and data plane traffic do not adversely affect the control plane. Without proper protection, it is possible for a data plane event, such as a DoS attack, to impact the control plane and cause disruption to the router. In a worst-case scenario, a data plane event could cause disruptions to the wider network. The following sections discuss Cisco IOS XR Software security features and configurations that can help protect and ensure the resilience of the control plane.

General Control Plane Hardening

Protection of the control plane of a network device is critical because the control plane ensures that the management and data planes are maintained and operational.
In many cases, disabling the reception and transmission of certain types of messages on an interface can minimize the amount of CPU load that is required to process unnecessary packets.

IP ICMP Redirects

An ICMP redirect message can be generated by a router when the following conditions are met:
The interface on which the packet enters the router is the same interface on which the packet is routed out.
The subnet or network of the source IP address is on the same subnet or network of the next-hop IP address of the routed packet.
The datagram is not source-routed (not a common occurrence).
If these conditions are met, the router forwards the packet to the next hop (by way of the same subnet using the same interface where the packet arrived) and sends an ICMP (Type 5) redirect message back to the sender of the original packet. This behavior allows the sender

to bypass the router and forward subsequent packets directly to the destination or to a router closer to the destination.
There are four types of ICMP redirect messages with the ICMP code 0-3. A malicious user can exploit the ability of the router to send ICMP redirects by continually sending packets to the router, forcing the router to respond with ICMP redirect messages that can have an adverse impact on the CPU and the performance of the router.
In Cisco IOS XR Software, ICMP redirects are disabled by default. To enable ICMP redirects, the **ipv4redirects** command should be used in the interface configuration mode. The use of this functionality should be controlled.

IP Direct-Broadcast

An IP directed-broadcast is an IP packet with the destination address of a particular IP subnet but that originates from a node that is not part of the destination subnet. Dropping IPv4 directed-broadcasts makes routers less susceptible to DoS attacks, especially the "smurf" attack. In such attacks, the attacker sends IP ICMP echo request from a source with a spoofed IP address to a directed-broadcast address. The request causes all the hosts on the targeted subnet to send replies to the spoofed IP source, causing a flood of ICMP response messages. This action could cause the exploited router to become unusable or consume a significant amount of bandwidth.
In a Cisco IOS XR device, directed-broadcast packets are dropped by default. To enable forwarding of IPv4 directed-broadcasts on an interface, use the **ipv4 directed-broadcast** command in interface configuration mode. If an interface requires IP directed-broadcast support, administrators are advised to control the use of the feature.

Disable IPv4 Source Routing
IPv4 source routing takes advantage of either the Loose Source Route with Record Route Layer 3 header options or the Strict Source Route with the Record Route header option to enable the source of the IP datagram to specify the network path that a packet will take. Attackers could use this functionality to bypass security controls in the network.
The Cisco IOS XR device discards (drops and does not process) by default all IPv4 datagrams that contain either the Loose or Strict Source Route header option. No configuration is required unless the default behavior needs to be changed.
Note: This default configuration is different from Cisco IOS Software in which IPv4 source route packets are processed by default.
The following configuration example illustrates the use of the global command to disable the previously enabled processing of source route packets:

```
!
no ipv4 source−route
!
```

ICMP Destination Unreachable

The ICMP Destination Unreachable error message is generated by the router to inform the source that the destination is unreachable for some reason. Generating these ICMP (Type 3) messages can increase CPU utilization on the device.

There are 16 types of ICMP destination unreachable messages with the ICMP code 0-15. Certain management functions take advantage of these error messages. For example, trace route relies on receiving the ICMP Port Unreachable (Type 3, code 3) message to determine when it has reached the last hop. Path MTU Discovery relies on the Datagram Too Big, Fragmentation Required Unreachable (Type 3, code 4) message to discover the appropriate Maximum Transmission Unit (MTU) size.
However, a malicious user could exploit the ability of the router to send ICMP unreachable by continually sending packets to the router. An exploit may force the router to respond with ICMP unreachable, resulting in an adverse impact on the CPU and the performance of the router. ICMP unreachable message generation is enabled by default and can be disabled using the interface configuration command

```
!
interface HundredGigE0/0/0/24
 ipv4 unreachable disable
!
```

When IP unreachable generation is enabled, the generation of these messages is rate-limited by default to one message per 500 milliseconds (ms). However, this default value can be changed by using the global configuration command as shown in the following example:

```
!
icmp ipv4 rate-limit unreachable [DF] < #of messages per ms
!
```

Cisco IOS XR Software maintains two timers: one for general destination unreachable messages and one for data fragmentation (DF) destination unreachable messages. Both timers share the same time limits and defaults. If the DF keyword is not configured, the **icmp ipv4 rate-limit unreachable** command sets the time values for DF destination unreachable messages. The DF option limits the rate at which ICMP destination unreachable messages are sent for Type 3, code 4 "packet-too-big, fragmentation is needed, and data fragmentation bit is set". If the DF option is configured, its time values remain independent from values of general destination unreachable messages. Without the DF option configuration, all messages would share the same time limits and defaults, as shown in the following configuration:

```
!
icmp ipv4 rate-limit unreachable 1000
!
```

IPv4 Options Packets

IPv4 packets that contain header options present two security concerns. First, such traffic must be process-switched by Cisco IOS XR devices, resulting in higher CPU usage (that is, the software path using the main CPU of the card is used instead of the hardware path forwarding). Second, packets with IPv4 source route header options can alter the path that traffic takes through the network, as previously described in the IPv4 Source Routing section. This behavior could allow such packets to subvert security controls.
In Cisco IOS XR Software, all IPv4 packets with any header option other than the "source-route" header options that are dropped by default (as detailed in the Disable IPv4 Source Routing section of this guide) are punted, police rate-limited, and processed by ucode and static policers depending on the destination address of the packets. In this way, Cisco IOS XR

software prevents packets with IPv4 header options from overwhelming or elevating the CPU load by default.
Note: The rate-limit value is not configurable.
The **show ipv4 traffic** EXEC command can be used to determine the number of IPv4 options packets that are received and processed.
See the Control Plane section of this guide for more information about LPTS and static policers.

IPv4 Fragments Rate Limiting

Fragmented IPv4 packets can pose a challenge to network devices. Fragmentation is often used in attempts to evade detection by ACLs, intrusion protection systems, and firewalls. Additionally, when the router is the destination of the fragmented IPv4 packet, these packets can be punted to the LC CPU and cause high CPU usage. For these reasons, IPv4 fragments are often used in DoS attacks.
In Cisco IOS XR Software versions 3.6 and later, configurable LPTS hardware policers are used to set the fragment packets default rate to limit the number of fragmented packets that can be punted to the LC CPU, which helps to reduce LC CPU usage.
The following configuration example illustrates the use of LPTS configurable policers:

```
!
lpts pifib hardware police
 flow fragment rate 10
!
```

Note: Administrators are advised to set IP fragment rates as low as possible. The current Cisco IOS XR device default fragment rate is 1000 p/s. This rate can cause significant LC CPU utilization when receiving a high volume of fragment packets. For Cisco IOS XR Software Release versions prior to 3.6, administrators should contact the Cisco Technical Assistance Center if default LPTS hardware police values require adjustment.

IPv4 Fragments Filtering

The filtering of fragmented IP packets can pose a challenge to security devices, because the Layer 4 information that is used to filter TCP and UDP packets is present only in the initial fragment.
Cisco IOS XR Software checks non-initial fragments of the packets by evaluating them against the ACL and ignoring any Layer 4 filtering information. This behavior causes non-initial fragments to be evaluated solely on the Layer 3 portion of any configured access control entry (ACE).
In the following example configuration, if a TCP packet destined for 192.168.1.1 on port 22 is fragmented in transit, the initial fragment is dropped as expected by the second ACE based on the Layer 4 information within the packet. However, all remaining (non-initial) fragments are allowed by the first ACE based on the Layer 3 information in the packet and ACE. *ipv4 access-list ACL-FRAGEMENT-EXAMPLE*

```
!
 ipv4 access-list ACL-FRAGEMENT-EXAMPLE
!
10 permit tcp any host 192.168.1.1 eq 80
```

20 deny tcp any host 192.168.1.1 eq 22
!
Due to the non-intuitive nature of fragment handling, IP fragments are often inadvertently permitted by ACLs. Fragmentation is also often used in attempts to evade detection by intrusion detection systems. It is for these reasons that IP fragments are often used in attacks, and why they must be explicitly filtered at the top of any configured ACLs. The following example ACL includes comprehensive filtering of IP fragments. The functionality in the following example must be used in conjunction with the functionality in the preceding example.
!
ipv4 access-list ACL-INFRASTRUCTRE-IN
!
!---Deny IP fragments using protocol-specific ACEs to aid in
!---classification of attack traffic
!
10 deny tcp any any fragments
20 deny upd any any fragments
30 deny icmp any any fragments
40 deny ipv4 any any fragements
!
!---Deny all other IP traffic to any network device
!
50 deny ipv4 any
!
!—Permit transit traffic
!
60 permit ipv4 any any
!
See the Access Control Lists and IP Fragments white paper for more details about advanced configuration.
See the Extended Access Control Lists with Fragment Control section of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information about ACL handling of fragmented IP packets.

IPv6 ICMP Rate Limiting

IPv6 ICMP error messages are generated by the router for many reasons. The generation of IPv6 ICMP error messages is enabled by default, but the rate at which these messages are generated can be rate-limited. This action is accomplished by using the global configuration command.
!
ipv6 icmp error-interval [milliseconds] [bucketsize]
!
The IPv6 ICMP rate limiting feature implements a token bucket algorithm and limits the rate at which IPv6 ICMP error messages are sent out on the network. The initial Cisco IOS XR Software implementation of IPv6 ICMP rate limiting defined a fixed interval between error messages. However, some applications, such as trace route, often require replies to a group

of requests that are sent in rapid succession. The fixed interval between error messages is not flexible enough to work with applications such as trace route and can cause the application to fail. Implementing a token bucket scheme allows several tokens—representing the ability to send one error message each—to be stored in a virtual bucket. The maximum number of tokens allowed in the bucket can be specified, and for every error message sent, one token is removed from the bucket. If a series of error messages is generated, error messages can be sent until the bucket is empty. When the bucket becomes depleted of tokens, IPv6 ICMP error messages will not be sent until a new token is placed in the bucket. The token bucket algorithm does not increase the average rate limiting time interval and is more flexible than the fixed time interval scheme. The following sample configuration shows how to change the interval time:

```
!
ipv6 icmp error-interval 50 20
!
```

IPv6 Packet with Header Extension

By default, Cisco IOS XR Software only punts to the CPU for handling IPv6 packets that contain the Hop-by-Hop header extension (NH=0). Currently, Cisco IOS XR software does not process IPv6 Routing Header Extensions (NH=43) and others. The default and only available mode of operation is to drop IPv6 packets with routing headers.

Note: Cisco IOS XR Software does not distinguish between Type 0 (source-route) and Type 2 (Mobile IP) Routing Header Extensions.

Punted IPv6 header extension packets are rate-limited by ucode in hardware. In this way, Cisco IOS XR Software prevents packets with IPv6 header extension from overwhelming or elevating the CPU load.

Note: The rate-limit value is not configurable.

TCP Service and Accept Rate Limiting

Most TCP services are disabled by default in Cisco IOS XR Software. After enabling a particular TCP service, a restriction of the incoming traffic rate should be enabled using MPP, peer control, or ACL.

Another way of restricting TCP traffic is to limit the number of accepted incoming TCP connections per second. Doing so can help mitigate DoS attacks from trusted neighbors and reduce CPU usage. The current default rate is 500 connections per second; this number should be carefully tuned on a case-by-case basis.

The following example shows how to reduce the number of accepted TCP connections to 200 connections per second:

```
!
tcp accept-rate 200
!
```

Proxy Address Resolution Protocol

Proxy address resolution protocol (ARP) is the technique in which one device, usually a router, answers ARP requests that are intended for another device. By acting on behalf of

the other device, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help networked devices on one subnet reach devices on remote subnets without configuring routing or a default gateway. Proxy ARP is defined in RFC 1027.
There are several disadvantages to using proxy ARP. The use of proxy ARP can result in increased ARP traffic on the network, segment and resource exhaustion, and man-in-the-middle attacks. Proxy ARP can be used by a resource exhaustion attack vector because each proxy ARP request consumes a small amount of memory. An attacker could exhaust all available memory by sending a large number of ARP requests. Man-in-the-middle attacks enable a host on the network to spoof the MAC address of the router, resulting in unsuspecting hosts sending traffic to the attacker.
The proxy ARP function is disabled by default in Cisco IOS XR software. To enable it, a **proxy-arp** command must be used on the interface configuration mode. If the proxy ARP function is absolutely required, administrators should control its use.

Network Time Protocol

Maintaining accurate time is essential for many aspects of network operations and security. The Network Time Protocol (NTP) is a UDP-based protocol that provides the ability to accurately maintain consistent time across many network devices. NTP is especially useful to ensure that timestamps on log messages are consistent throughout the entire network. To prevent security attacks, NTP protocol should always use trusted time sources and proper authentication.
Two mechanisms for securing NTP are available: an access list-based restriction scheme and an encrypted authentication mechanism.
Access list-based restriction: The access list-based restriction scheme allows administrators to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet. The following example shows access list-based restriction using access group:

```
(config)# ntp
(config-ntp)# peer 10.1.1.1
(config-ntp)# peer 10.2.2.2
(config-ntp)# access-group peer peer-acl
(config-ntp)# access-group serve serve-acl
(config-ntp)# access-group serve-only serve-only-acl
(config-ntp)# access-group query-only query-only-acl
```

NTP Authentication: NTP authentication configuration provides the assurance that all NTP messages are exchanged between trusted NTP peers and that their content has not been modified. The following example shows an NTP authentication configuration:
NTP Client

```
(config)# ntp
(config-ntp)# authenticate
(config-ntp)# authentication-key 5 md5 encrypted ciscotime
(config-ntp)# trusted-key 5
(config-ntp)# server 172.16.1.5 key 5
```

NTP Server

(config)# ntp
(config-ntp)# authenticate
(config-ntp)# authentication-key 5 md5 encrypted ciscotime
(config-ntp)# trusted-key 5
Note: The encryption and decryption processes used in NTP authentication can be very CPU-intensive and can seriously degrade the accuracy of the time that is propagated within a network. On networks that permit a more comprehensive model of access control, administrators should consider using the access-list-based form of control instead.
NTP services are disabled on all interfaces in Cisco IOS XR Software by default. Administrators should enable it only on the specific interface when necessary. When NTP is enabled globally, administrators can selectively prevent NTP packets from being received through a specific interface by turning off NTP on a given interface, as shown in the following example:
(config)# ntp
(config-ntp)# no interface HundredGigE 0/0/0/1
or
(config)# ntp
(config-ntp)# interface HundredGigE 0/0/0/1 disable

Limit CPU Impact of Control Plane Traffic

Protection of the control plane is critical. Because application performance and end-user experience can suffer without the presence of data and management traffic, the survivability of the control plane ensures that the other two planes are maintained and operational.

Control Plane Traffic

Under normal network operating conditions, most packets handled by network devices are data plane packets, and routers are optimized to forward these packets efficiently in the hardware and fast paths. However, there are certain types of packets that must be punted to be handled directly by the LC or RP CPUs. The types of packets in this group can be divided into the following categories.
Note: Packets that are listed in *italics* include traffic that is punted to RP CPU. The packets listed in straight text are processed directly by the LC CPU.
Transit traffic: IP option, *IP option router alert*, ARP resolution, and TTL expired packets
Unicast receive (to us) traffic: Various ICMP, *management traffic like* SSH*, SNMP and XML*, NetFlow, and Cisco DP packets along with routing packets such as *OSPF, BGP,* and *ISIS*
Multicast and Broadcast: Multicast control traffic like *PIM* and *HSRP*, broadcast packets, and the first packet of MCAST stream
Special Traffic: IP and MPLS fragments and L2 packets
The preceding traffic types can be a potential source of DoS attacks and the system must have the ability to police such flows.

Local Packet Transport Services

As routers handle greater traffic loads and networks become larger and more complex, it is increasingly difficult for network engineers to manually configure all the controls necessary

to protect the router and core network infrastructure. The Cisco IOS XR Software directs router self-protection beyond manual configuration and the mere detection and reporting on security violations. Cisco IOS XR Software provides intelligence that automatically provisions many aspects of router self-protection functions, including differentiating between unidentified and trusted control plane sessions. In a Cisco IOS XR device, this functionality is provided by hardware rate-limiters that are associated with LPTS. The primary functions of LPTS include the following attributes:
Ability to control packet flows for all internal applications that receive packets from outside the router.
Exceptions packet rate-limiters that control all other packets that require punting for CPU support.
Preconfigured LPTS and exceptions packet rate-limiters for use with default values; these rate-limiters are capable of functioning without the need for any user-configuration.
Static programming of LPTS policers during boot; these policers are automatically applied based on the flow type of the incoming traffic. This automation frees time spent by network administrators on manual configuration for use on other mission-critical tasks.
User capability to configure customized rate-limits if the default policer settings are inadequate for the specific deployment model. The user can configure the rate per policer per node (locally) or globally from CLI, thus overwriting default static policer values. This attribute applies to Cisco IOS XR Software Release versions 3.6 and later.
The command that is used to configure LPTS police for a particular flow type is as follows:
[no] lpts pifib hardware police [location]
If the "location" option is not specified, it is equivalent to the "location all" option in the configuration command, causing the command to be applied on all nodes.
For example, BGP-known traffic is controlled in the following configuration:
!
lpts pifib hardware police
flow bgp known rate 20000
!
Default LPTS policer values and flow-type values can be seen using the
following **show** command:
!
show lpts pifib hardware police [location {all | }]
!
In addition to LPTS, which controls the flow of all receive packets, a separate group of hardware-based rate-limiters automatically control "exceptions" along with transit packets that require punting for CPU support. Examples of packets of this type include various Layer 2 packets, Cisco DP, IPv4 TTL errors, IPv4 options, ARP and RARP, and many others. Because these packet types do not rep resent traffic flows that the router wants to keep track of, there is no need to provide per flow control, which is the case with LPTS and "receive" packets. Therefore, these "exceptions" rate-limiters are simply programmed statically and are not user-configurable.
See the Implementing LPTS section of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information.

CPU and Routing Protocol Software Queues

When properly configured, LPTS policers should mitigate most cases where overflowing the line card (LC) or the routing protocol (RP) CPU may occur. Cisco IOS XR Software implements additional mechanisms in case the amount of punted packets exceeds the processing capability of the CPU. Before the packets reach a CPU, they are placed in one of several software queues that perform priority queuing should congestion occur.
Four queues that classify packets are available for LC CPU as shown in the following:
Critical: Layer 2 keep alive (PPP, HDLC)
High: ARP
Medium: IPv4 lookup
Low: TTL errors, Options, logging, ICMP
Three queues that classify packets are available for the RP as seen in the following list:
High: OSPF, ISIS
Medium: BGP, PIM, LDP, SSH
Low: Fragments

General Routing Protocol Securing Techniques

Multiple Cisco IOS XR features have been developed to protect IP protocols against various attacks. The following section details protocol-independent features. The sections that follow will focus on the specific examples of the protocol implementation security features.

Message-Digest Algorithm 5 Peer Authentication (Take it out)

Message-Digest algorithm 5 (MD5) is a widely used cryptographic hash function with a 128-bit hash value. Most routing protocols can use MD5 to generate a hash-based MAC. MAC provides an integrity check to protect control packets that are exchanged between routing protocol session peers. Each protocol packet carries an MD5 digest message. This digest acts like a signature for that segment, incorporating information that is known only to the connection end points. Using MD5 authentication significantly mitigates certain types of attacks against routing protocols. MD5 should be configured on each session. Protocols like BGP, OSPF, ISIS, and LDP have the option to configure MD5 to secure their sessions.

Keychain Management

Routing protocols and network management applications often use authentication for enhanced security while communicating with their peers. Most authentication methods use shared secrets, sometimes referred to as keys, on all the entities in their mechanisms for establishing trust between each peer.
As with all key-based security methods, it is useful to specify a lifetime for each key so that these keys are changed on a regular basis when they expire. To maintain stability during such key changes, each party must be able to store and use more than one key for an application at the same time. The sequence of keys intended for authenticating the same peer or peer group is collectively managed in a container called a key chain. Each key in the key chain has an associated lifetime. Multiple cryptographic algorithms options can also be specified.
A keychain is defined as a set of keys that each include an associated lifetime, authentication-algorithm, and key-id. Hitless key rollover provides a switchover from one (expired) key to another (active) key, without any session-drops. Border Gateway Protocol

(BGP), Open Shortest Path First (OSPF), and Intermediate System-to-Intermediate System (IS-IS) use the keychain to implement a hitless key rollover for authentication.
The following is an example of the keychain configuration:

```
!
key chain my-key-chain
 accept-tolerance 10000
 key 8
  key-string mykey_test
  cryptographic-algorithm AES-128-CMAC-96 [MD5][SHA-1]
  send-lifetime 1:00:00 june 29 2021 infinite
  accept-lifetime 1:00:00 june 29 2021 infinite
!
```

See the Implementing Keychain Management section of the Cisco IOS XR System Security Configuration Guide for more information about keychain management.

Routing Policy Language

In Cisco IOS XR Software, a new routing policy language (RPL) has been designed to provide a single, straightforward language in which all routing policy needs can be expressed. RPL was designed to support large-scale routing configurations and thus can be used by BGP and by IGP protocols such as OSPF and ISIS. RPL replaces the route map statements that are used in Cisco IOS Software.
The policy language introduces the notion of route policies and sets. Sets are containers of similar data that can be used in route attribute matching and setting operations. Four set types exist: prefix-sets, community-sets, as-path-sets, and ext community-sets. These sets hold groupings of IPv4 or IPv6 prefixes, community values, autonomous system (AS) path regular expressions, and extended community values, respectively.
Route policies use the sets to instruct the router to inspect and filter routes and potentially modify route attributes when they are accepted from a peer, advertised to a peer, or redistributed from one routing protocol to another. Routing protocols make decisions to advertise, aggregate, discard, distribute, export, hold, import, redistribute, and otherwise modify routes based on configured routing policy.
Route policies can be attached to different attach points when an association is formed between a specific protocol entity, in this case, a BGP neighbor, and a specific named policy.
RPL is a very powerful tool and this document is not intended to describe the whole language in detail. For more information about RPL, see the Implementing Routing Policy section of the Cisco IOS XR Routing Configuration Guide.

Secure Border Gateway Protocol

The Border Gateway Protocol (BGP) is the routing foundation of the Internet. As such, any organization with more than modest connectivity requirements often finds itself utilizing BGP. BGP is often targeted by attackers because of its ubiquity and the "set and forget" nature of BGP configurations in smaller organizations. However, there are many BGP security features that can be leveraged to increase the security of a BGP protocol.
This section provides an overview of the most important BGP security features. Where appropriate, configuration recommendations are made.

See the Implementing BGP section of the Cisco IOS XR Routing Configuration Guide for more information.

BGP Time-to-Live-Based Security Protection

The BGP time-to-live (TTL) -based security check is designed to protect BGP processes from CPU utilization-based attacks. The BGP protocol defines two types of sessions: "internal" BGP sessions (iBGP) that are established between peers within the same Autonomous System (AS), and external BGP (eBGP) sessions that are established between peers in two different ASs. Of particular interest are the eBGP sessions, which are the type of BGP sessions that would be established between an Enterprise and its upstream service provider. By default, and per RFC 3682, when eBGP is configured, the IP header TTL for all neighbor session packets is set to "1". This configuration was originally assumed to be useful because it prevented the establishment of an eBGP session beyond a single hop; however, the configuration does not consider an attacker who could be located up to 255 hops away and could have the ability to send spoofed packets to BGP-speaking routers. For example, an attacker could send large amounts of TCP SYN packets at the BGP peer to overwhelm the BGP process. The BGP TTL security check leverages ISP eBGP peering sessions between routers that are adjacent to each other (either between directly connected interfaces or possibly between loopbacks). Because TTL spoofing is considered nearly impossible, a mechanism that is based on an expected TTL value was developed to provide a simple yet robust defense from infrastructure attacks that are based on forged BGP packets.
Note: This concept was originally defined in the The BGP TTL Security Hack (BTSH) document and later extended beyond BGP in RFC 3682, The Generalized TTL Security Mechanism (GTSM).
When the BGP TTL security check is enabled, the initial value for eBGP is set to "255" (instead of "1"), and a minimum TTL value is enforced on all BGP packets that are associated with that eBGP session. Because the IP header TTL value is decremented by each hop (router interface) along its path to its final destination, using BTSH restricts possible attack origins to the directly connected routers.
Currently, Cisco IOS XR Software does not support GTSM for eBGP multi-hop scenarios; only directly connected or loopback-based eBGP peering sessions can use the GTSM feature. This feature is checked in hardware during packets ingress in LCs. GTSM for Cisco IOS XR BGP can be enabled using the following commands:

```
!
router bgp 100
 neighbor 1.2.3.4
  remote-as 123
  ttl-security
!
```

BGP Peer Authentication with MD5

BGP neighbor sessions are established between two peers and then routes are exchanged between each other. By enabling the MD5-based neighbor authentication mechanism, administrators can ensure that only authorized peers establish this BGP neighbor

relationship, and that the routing information exchanged between these two devices has not been altered en-route between the two systems.
The BGP neighbor authentication process is a symmetric technique. Therefore, when this process is turned on, it must be simultaneously enabled on both sides of the peering session. Neighbor authentication using MD5 creates an MD5 digest hash for each packet that is sent as part of a BGP session. Specifically, portions of the IP and TCP headers, TCP payload that includes the BGP route advertisements, and the shared secret key, are used to generate the digestMD5 hash by the sending router. The created digest MD5 hash is then stored in TCP option Kind 19, which was created specifically for this purpose in the RFC 2385. The receiving BGP speaker neighbor uses the same algorithm and shared secret key to regenerate and compute its own version of the message digestMD5 hash. The receiving BGP speaker neighbor compares its own version with the one it received; if the received and computed digestsMD5 hash values are not identical, the packet is discarded. Otherwise, the packet is accepted and processed by BGP.
Peer authentication with MD5 is configured by using the password option to the neighbor BGP router configuration command. The following example illustrates the use of this command:

```
!
router bgp 100
 neighbor 1.2.3.4
  remote-as 123
  password {clear | encrypted} password
!
```

See RFC 2385, Protection of BGP Sessions via the TCP MD5 Signature Option, for more information about BGP MD5 peering authentication.

BGP Peer Authentication with Keychain

BGP uses TCP authentication, which enables the authentication option and sends the MAC based on the cryptographic algorithm configured for the keychain.
The following example shows how to apply the keychain between BGP neighbors:

```
!
router bgp 100
 neighbor 1.2.3.4
  remote-as 123
  keychain my-key-chain
!
```

BGP Maximum Prefixes

BGP prefixes are stored by a router in memory. The more prefixes a router must hold, the greater the memory consumed by BGP. Both malicious and inadvertent (misconfiguration) processes can cause excessive prefixes to be seen by a BGP peer. To prevent memory exhaustion, it is important to configure the maximum number of prefixes accepted on a per-peer basis. Administrators are advised to configure limits for each BGP peer.
When configuring BGP prefixes using the neighbor maximum-prefix BGP router configuration command, one argument is required: the maximum number of prefixes that are accepted

before a peer is shutdown. Optionally, a number from 1–100 can also be entered. This number represents the percentage of the maximum prefixes value at which point a log message can be sent. The following is an example of a BGP configuration:

```
!
router bgp 100
 neighbor 1.2.3.4
  remote-as 123
 address-family   unicast
  maximum-prefix  [ipv4}[ipv6] [threshold] [warning-only]
!
```

The BGP maximum-prefix feature allows users to control the number of prefixes that can be installed from a neighbor. When configured, this feature will bring down a neighbor relationship when the number of received prefixes from that peer exceeds the configured maximum-prefix limit. Typically, the maximum-prefix threshold would be set with some margin of error over a target value to accommodate some degree of unexpected changes in topology. The other warning-only option does not cause the neighbor relationship to be brought down, but simply issues a message when the threshold is exceeded. This feature is commonly used for external BGP peers but can also be applied to internal BGP peers.

Filter BGP Prefixes with RPL policies

In Cisco IOS XR devices, BGP allows users to filter prefixes that are based on network prefixes, as-paths, and community or ext-community values.

Prefix sets allow a network administrator to permit or deny specific prefixes that are sent or received by way of BGP. Prefix sets should be used when possible, to ensure that network traffic is sent over the intended paths. Prefix lists should be applied to each eBGP peer in both the inbound and outbound directions.

Configured prefix sets limit the prefixes that are sent or received to prefixes specifically permitted by the routing policy of a network. If this is not feasible because of many prefixes received, a prefix set should be configured to specifically block known bad prefixes. These known bad prefixes include unallocated IP address space and networks that are reserved for internal or testing purposes by RFC 3330. Outbound prefix lists should be configured to specifically permit only the prefixes that an organization intends to advertise.

The following is a simple example of the BGP route policy using a prefix set to drop 10.0.2.0/24 and 10.0.3.0/24 incoming prefixes. It uses *prefix-set* and the policy is attached to the neighbor inbound attach point.

```
!
prefix-set bgp_route_drop
 10.0.2.0/24,
 10.0.3.0/24
end-set
!
route-policy bgp_in
 if destination in bgp_route_drop then
  drop
 endif
end-policy
```

```
!
router bgp < as-number >
 neighbor < address >
  address-family ipv4 unicast
   route-policy bgp_in in
!
```

The following configuration uses *as-path-set* to permit inbound prefixes, which include AS path 12 or 13, but to drop other prefixes:

```
!
as-path-set my-as-set
 ios-regex '12$',
 ios-regex '13$'
end-set
route-policy policy-a
 if as-path in my-as-path then
  pass
 else
  drop
 endif
end-policy
!
router bgp < as-number >
 neighbor < address >
 address-family ipv4 unicast
  route-policy policy-a in
!
```

For more information about RPL BGP, see the Implementing Routing Policy section of the Cisco IOS XR Routing Configuration Guide.

Secure Interior Gateway Protocols

The ability of a network to properly forward traffic and recover from topology changes or faults is dependent on an accurate view of the topology. Running an IGP can often provide this view. By default, IGPs are dynamic and discover additional routers that communicate with the particular IGP in use. IGPs also discover routes that can be used during a network link failure.

The following subsections provide an overview of the most important IGP security features. Recommendations and examples that cover OSPF, ISIS, Routing Information Protocol Version 2 (RIPv2), and Enhanced Interior Gateway Routing Protocol (EIGRP) are provided when appropriate.

See the Cisco IOS XR Routing Configuration Guide for more information about IGP protocol configuration.

IGP TTL Security

The Generalized TTL Security Mechanism (GTSM) (RFC 3682) is designed to protect the control plane of a router from CPU utilization-based attacks. GTSM recognizes that the vast majority of protocol peerings are established between routers that are adjacent to or, in a worst case scenario, between loopbacks on adjacent routers. Because TTL spoofing is considered nearly impossible, a mechanism that is based on an expected TTL value can provide a simple and reasonably robust defense from infrastructure attacks that are based on forged protocol packets. GTSM is equally applicable to both TTL (IPv4) and Hop Limit (IPv6).
The initial Cisco IOS XR Software implementation for GTSM supported BGP peering protection; see the section of this guide for more information. Cisco IOS XR Software provides extended support for GTSM to cover the OSPF IGP.
OSPF is a link state protocol that requires networking devices to detect topological changes in the network; flood Link State Advertisement (LSA) updates to neighbors; and quickly converge on a new view of the topology. However, while receiving LSAs from neighbors, network attacks can occur, because no checks occur to distinguish whether unicast or multicast packets are originating from a neighbor that is one hop away or multiple hops away over virtual links. For virtual links, OSPF packets travel multiple hops across the network; therefore, the TTL value can be decremented several times. For virtual links, a minimum TTL value must be allowed and accepted for multiple hop packets.
The following is a sample configuration of GTSM for OSPF. GTSM can be configured on a per-interface basis or globally for all OSPF peers. The number of hops is the parameter that changes the default expected TTL value of incoming packets.

```
!
router ospf 1
 area 1
  interface HundredGigE0/0/0/23
   security ttl hops 2
!
```

IGP Peer Authentication with MD5

Failure to secure the exchange of routing information could allow an attacker to introduce false routing information into the network. By using password authentication with routing protocols between routers, administrators can help secure the network. Because this authentication is sent as plain text; however, the attacker could easily subvert this security control. By adding MD5 hash capabilities to the authentication process, the routing updates no longer contain plaintext passwords, and the entire contents of the routing update are more resistant to tampering.
MD5 authentication is still susceptible to brute force and dictionary attacks if weak passwords are chosen. Administrators are advised to use passwords with sufficient randomization. MD5 authentication is more secure than password authentication
The following subsections detail the specific implementations of the OSPF and IS-IS protocols.

OSPF Authentication with MD5

All OSPF routing protocol exchanges are authenticated, and the method used can vary depending on how authentication is configured. When using cryptographic authentication, the OSPF routing protocol uses the MD5 authentication algorithm to authenticate packets that are transmitted between neighbors in the network. For each OSPF protocol packet, a key is used to generate and verify a message digest that is appended to the end of the OSPF packet. The message digest is a one-way function of the OSPF protocol packet and the secret key. Each key is identified by the combination of the interface used and the key identification.
The following configuration example shows OSPF authentication with MD5 Configuration:

```
!
router ospf 1
 area 1
  authentication message-digest
   message-digest-key 100 md5 cisco
!
```

Intermediate System to Intermediate System Authentication with MD5

Authentication can be configured to limit the establishment of adjacencies by using the hello-password command and limit the exchange of LSPs by using the **lsp-password** command.
Intermediate System to Intermediate System (IS-IS) supports plaintext authentication, which does not provide security against unauthorized users. The password is exchanged as plain text and is potentially visible to an agent who can view the IS-IS packets. When an HMAC-MD5 password is configured, the password is never sent over the network and, instead, is used to calculate a cryptographic checksum to ensure the integrity of the exchanged data.
To set the domain password, configure the lsp-password command for Level 2; to set the area password, configure the lsp-password command for Level 1.
The following configuration example shows IS-IS authentication with MD5 configuration:

```
!
router isis Core
 lsp-password hmac-md5 cisco123
 interface HundredGigE 0/0/0/23
  hello-password hmac-md5 cisco123
!
```

Open Shortest Path First Authentication with Keychain

The following examples show the OSPF configuration, which establishes a keychain to set the password at the global, area, and interface levels.
Router-level configuration:

```
!
router ospf 1
 router-id 10.2.3.4
 authentication message-digest keychain my-key-chain
!
```

Area-level configuration:

```
!
router ospf 1
 router-id 10.2.3.4
 area 0
  authentication message-digest keychain my-key-chain
!
Configuration at virtual-link level:
!
router ospf 1
 router-id 10.2.3.4
 area 0
 virtual-link 1.1.1.1
authentication message-digest keychain my-key-chain
!
Interface-level configuration:
!
router ospf 1
 router-id 10.2.3.4
 area 0
 interface
authentication message-digest keychain my-key-chain
!
```

IS-IS Authentication with Keychain

The following example shows IS-IS configuration, which configures a keychain to set the domain and area passwords at the global and interface levels.

```
!
router isis 1
 lsp-password keychain isis-keys level-1
 interface
hello-password keychain isis-keys
!
```

Passive Interface

The use of the passive interface capabilities within IGPs has two benefits. First, such capabilities reduce the CPU load on the router by suppressing the generation and processing of IGP protocol messages on the identified interface(s). Second, these capabilities could help mitigate information leaks by not advising IGP protocol information to networks beyond administrative control.

The following example shows the use of the passive command for the OSPF protocol. If the passive command is not configured, the interface generates and receives normal OSPF traffic flow (default behavior).

```
!
router ospf 1
 area 0
```

```
 interface Loopback0
  passive
!
```
The following shows the ISIS ISIS configuration example:
```
!
router isis 1
 interface Loopback0
  passive
!
```

IGP Route Filtering

Like BGP, the IGP protocols also benefit from having the ability to filter ingress and egress routing information and can use the Cisco IOS XR Software RPL language for this purpose. See the Implementing Routing Policy section of the Cisco IOS XR Routing Configuration Guide for more information.

IGP Routing Process Resource Consumption

Routing Protocol prefixes are stored by a router in memory, and resource consumption increases with the additional prefixes that a router must hold. To prevent resource exhaustion, it is important to set protocol limits for number of accepted and stored prefixes. See the Implementing Routing Policy section of the Cisco IOS XR Routing Configuration Guide for more information.

OSPF Maximum Redistributed Prefix Limit

To limit the aggregate number of routes that may be redistributed into an OSPF process, administrators are advised to use the **maximum redistributed-prefix** command in the appropriate mode. The following example shows how to configure a maximum number of routes that can be redistributed for an OSPF routing process:
```
!
router ospf 10
 maximum redistributed-prefixes 15000
!
```

IS-IS Redistributed Prefix Limit

To specify an upper limit on the number of external prefixes (subject to summarization) that the Intermediate System-to-Intermediate System (IS-IS) protocol advertises, use the **maximum-redistributed-prefixes** command in address family mode. The following example shows how to specify the number of redistributed prefixes at 5000 for Level 2:
```
!
router isis isp
 address-family ipv4 unicast
maximum-redistributed-prefixes 5000 level 2
!
```

Enhanced Interior Gateway Routing Protocol Redistributed Prefix Limit

To limit the number of prefixes that are redistributed to an Enhanced Interior Gateway Routing Protocol (EIGRP) process, use the **redistribute maximum-prefix** command in the IPv4 virtual routing and forwarding (VRF) address family configuration mode. The following example shows how to configure the maximum prefix limit for routes that are learned through redistribution under the **vrf vpn-1** command. The maximum limit is set to 5000 prefixes and the warning threshold is set to 95 percent. When the number of prefixes that is learned through redistribution reaches 4750 (95 percent of 5000), warning messages are displayed in the console. Because the warning-only keyword is configured, the topology and routing tables are not cleared, and route redistribution is not placed in a penalty state.

```
!
router eigrp 100
 vrf vpn-1
  address-family ipv4
   redistribute maximum-prefix 5000 95 warning-only
!
```

Secure Label Distribution Protocol

Label Distribution Protocol (LDP) performs label distribution in MPLS environments. As LDP assigns and distributes labels, the stability of this protocol is a critical factor for reliable MPLS networks. LDP uses the hello discovery mechanism to discover its neighbors and create sessions between them. To secure data transmitted over these sessions, password authentication (using the TCP MD5 option) should be configured for a given neighbor using password command as shown below:

```
!
mpls ldp
  …
neighbor  password clear cisco123 (password will get encrypted)
!
```

Secure Resource Reservation Protocol

The Resource Reservation Protocol (RSVP), as described in RFC 2205, is a Transport layer protocol that is designed to reserve resources across the network for integrated services. RSVP provides a receiver-initiated setup of resource reservations for multicast or unicast data flows. An extension of RSVP for traffic engineering is used for the MPLS TE tunnels setup.

Secure RSVP with ACL

Cisco IOS XR implementation of RSVP enables ACLs to forward, drop, or perform processing on RSVP Router-Alert (RA) packets. For each incoming RSVP RA packet, RSVP inspects the IP header and attempts to match the source and destination IP addresses with a prefix that is configured in an extended ACL. In the following example, rsvp_acl_example ACL is

configured to permit RA packets with the source address 1.1.1.1 and drop packets with the source address 2.2.2.2. All other RA packets are processed as normal RSVP packets.

```
!
ipv4 access-list rsvp_acl_example
 10 permit ipv4 host  1.1.1.1 any
 20 deny ipv4 any host 2.2.2.2
 !
rsvp
 signalling prefix-filtering access-list rsvp_acl_example
!
```

The default behavior of Cisco IOS XR Software will perform normal RSVP processing on RA packets when the ACL match returns an implicit (default) deny. To change the default behavior, the following command must be issued:

```
!
signalling prefix-filtering default-deny-action drop
!
```

RSVP Authentication

RSVP authentication supports only keyed-hash message authentication code (HMAC) type algorithms. The RSVP authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signaling messages digitally, thus allowing the receiver of an RSVP message to verify the sender of the message without relying solely on the IP address of the sender. The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message, as defined in RFC 2747.

RSVP authentication can be configured in global, interface, or neighbor configuration modes. In global mode, a single common key set is expected to be used to authenticate all RSVP messages, while in the other modes, different keys can be used. A keychain has to be configured first to enable authentication on the RSVP. See the Keychain Management section of this guide for more details about keychain creation. The following configuration shows how to enable RSVP authentications on different modes:

```
!
rsvp
 authentication
  key-source key-chain default_keys
 !
interface
 authentication HundredGigE 0/0/0/23
  key-source key-chain default_keys
  !
neighbor
 authentication
  key-source key-chain my-key-chain
!
```

Secure First Hop Redundancy Protocols

First Hop Redundancy Protocols (FHRPs) provide resiliency and redundancy for devices that are acting as default gateways. This situation and these protocols are commonplace in environments where a pair of Layer 3 devices provides default gateway functionality for a network segment or set of VLANs that contain servers or workstations.
The Hot Standby Router Protocol (HSRP) and the Virtual Router Redundancy Protocol (VRRP) are examples of FHRPs. By default, these protocols communicate using unauthenticated packets. This kind of communication could allow an attacker to pose as an FHRP-speaking device to assume the default gateway role on the network. This takeover would allow an attacker to perform a man-in-the-middle attack and intercept all user traffic that exits the network.

VRRP Text Authentication

Text authentication can ensure that VRRP messages received from adjacent routers that comprise a virtual router are authenticated by configuring a simple text password, as shown in the following configuration:

```
!
router vrrp
 interface HundredGigE0/0/0/23
  address-family ipv4
   vrrp 1
    text-authentication cisco123
!
```

HSRP Text Authentication

Like VRRP, HSRP authentication can be configured using the following authentication command:

```
!
router hsrp
 interface HundredGigE0/0/0/23
  address-family ipv4
   hsrp 1
    authentication cisco123
   !
```

See the Implementing HSRP and Implementing VRRP sections of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information about HSRP and VRRP.

Data Plane

The data plane contains the logical group of "customer" applications: traffic generated by hosts, clients, servers; and applications that are sourced from and destined to other similar devices supported by the network. Data plane traffic is typically forwarded in the fast path, although certain exception packets can be punted, requiring CPU assistance. Within the context of security and because the data plane represents the highest traffic rates, it is

critical that the data plane be secured to prevent exception packets from punting to the CPU and impacting the control and management planes.
Within the data plane, there are many features and configuration options that can help secure traffic. The following sections detail these features and options that can be implemented to further secure the network.

Filter Transit Traffic with ACLs

Administrators can control what traffic transits the network by using transit access control lists (tACLs). The tACLs contrast with the infrastructure ACLs that seek to filter traffic that is destined to the network device. The filtering provided by tACLs is beneficial when administrators want to filter traffic to a particular group of devices or traffic that is transiting the network.
tACLs are also an appropriate mechanism with which to implement static anti-spoofing protections. Refer to the Anti-Spoofing Protections section of this guide and the Implementing Access Lists and Prefix Lists section of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information.

IPv4 and IPv6 ACL with Counters

In Cisco IOS XR Software, ACL counters are maintained in hardware and software. Hardware counters are used for packet-filtering applications, such as when an access group is applied to an interface. Software counters are used by all the applications and mainly pertain to software packet processing.
To display the hardware counters for an access group, use the following command:
show {ipv4 | ipv6} access-lists
To clear the hardware counters, use the following command:
clear {ipv4 | ipv6} access-list
Hardware counters are disabled by default for IPv4 ACLs. To enable hardware counting, use the following command:
ipv4 access-group access-list-name {in | out} [hw-count]
Hardware counters are enabled only on the specified interface.
Software counters are updated only for the packets that are software processes—for example, exception packets that are punted to the LC CPU for processing, or an ACL that is used by routing protocols. This information also applies to IPv6 ACLs with one exception: hardware counting is always enabled for IPv6. Note that the *hw-count* option does not exist in the IPv6 access-group mode .

IPv6 ACL

In Cisco IOS XR Software, every IPv6 ACL has the following implicit statements as its last match entries:
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any

The first two match conditions allow for ICMPv6 neighbor discovery. The IPv6 neighbor discovery process makes use of the IPv6 network layer service; therefore, by default, IPv6 ACLs implicitly allow IPv6 neighbor discovery packets to be sent and received on an interface.
An IPv6 ACL must contain at least one entry for the implicit *deny ipv6 any* statement to take effect.

ICMP Packet Filtering

Certain ICMP message types are commonly used by network troubleshooting tools such as ping and trace route, as well as by Path MTU Discovery. There are other legitimate uses for ICMP. However, not all ICMP types are required for proper network operation, and ICMP flooding attacks can cause high CPU usage. It is necessary to have mechanisms that control which ICMP packet types are permitted to enter the network.
Cisco IOS XR Software provides ACL functionality to specifically filter ICMP messages by name or type and code. The following example shows IPv4 ACL allowing ICMP from trusted networks and permitting necessary ICMP type packets from other resources, while blocking all other ICMP type packets from other sources:

```
Ipv4 access-list ACL-TRANSIT-IN
!!--- Permit ICMP packets from trusted networks only !
10 permit icmp host x.x.x.x
!!--- Permit other ICMP traffic, if required, for proper Customer !
!!--- network operations !
20 permit icmp any any echo
30 permit icmp any any echo-reply
40 permit icmp any any ttl-exceeded
50 permit icmp any any unreachable
60 permit icmp any any packet-too-big
!!--- Deny all other IP traffic to any network device !
70 deny icmp any any
```

In the preceding example, 20 and 30 are for ping, 40 and 50 are for trace route, and 60 is for PMTUD.
Note: If the destination IP address happens to be receive, the preceding example is still relevant because Cisco IOS XR Software has LPTS to control packets that reach the CPU. As a result, there is a minimum risk of exhausting CPU resources. To prevent this risk on Cisco IOS devices, administrators should configure Control Plane Policing (CoPP).

Filter IPv4 Traffic with Remote Triggered Black Hole Filtering

Remote Triggered Black Hole Filtering uses the CEF process (fast path) to drop undesirable IPv4 traffic before it is forwarded to the network.
The following example provides a basic remote triggered black hole filtering configuration on an edge router:

```
community-set ddos-comm
1:666
end-set
!
```

```
route-policy RTBH-drop
if community matches-any ddos-comm then
set next-hop 192.0.2.1
set local-preference 200
endif
end-policy
!
router static
address-family ipv4 unicast
192.0.2.1/32 Null0
!
router bgp 65000
neighbor 191.0.0.2
remote-as 65188
route-policy RTBH-drop in
!
```

Note: In Cisco IOS XR Software, the Null 0 interface always exists and it never sends ICMP unreachable messages.

Filter IPv6 Traffic with Remote Triggered Black Hole Filtering

Remote Triggered Black Hole Filtering uses the CEF process (fast path) to drop undesirable IPv6 traffic before forwarding that traffic to the network.
The following example provides a basic remote triggered black hole filtering configuration on an Edge Router:

```
community-set ddos-comm
1:666
end-set
!
route-policy RTBH-drop-v6
if community matches-any ddos-comm then
set next-hop 2001:db8:bad::1
endif
end-policy
!
router static
address-family ipv6 unicast
2001:db8:bad::/48 Null0
!
router bgp 65000
neighbor 191.0.0.2
address-family ipv6 unicast
route-policy RTBH-drop-v6 in
!
```

Anti-Spoofing Protections

Many attacks utilize source IP address spoofing to conceal the true source of an attack and hinder accurate trace back or to defeat access control lists (ACLs). Cisco IOS XR Software provides the Unicast Reverse Path Forwarding (RPF) feature to deter attacks that rely on source IP address spoofing. In addition, ACLs and null routing are often deployed to augment spoofing protection.

Unicast Reverse Path Forwarding

Unicast Reverse Path Forwarding (Unicast RPF) provides source network verification and can reduce spoofed attacks from networks that are not under direct administrative control. Unicast RPF enables a device to verify that the source address of a forwarded packet can be reached through the interface that received the packet. Administrators must not rely on Unicast RPF as the only protection against spoofing. Spoofed packets can enter the network through a Unicast RPF-enabled interface if an appropriate return route to the source IP address exists. Unicast RPF relies on the Cisco Express Forwarding feature on each device and is configured on a per-interface basis.
Unicast RPF can be configured in either of the following modes:
Strict mode Unicast RPF: Strict mode checks the forwarding information base (FIB) to match the source address with the next hop adjacency. If the source address of the incoming packet does not match the next hop entry in the FIB for this address, the packet is dropped.
Loose mode Unicast RPF: Loose mode searches for the source address of a packet in the FIB table. If the address exists and matches a real and valid forwarding entry (not necessarily pointing to the ingress interface on which the packet was received), then the packet is further processed, otherwise it is dropped.
In cases where there is asymmetric routing, loose mode is preferred because strict mode is known to drop packets. During configuration of the ipv4 verify interface configuration command, the keyword *any* configures loose mode, while the keyword *rx* configures strict mode.
The following example illustrates configuration of the **ipv4 verify** command:
interface
ipv4 verify unicast source reachable-via rx
or
ipv4 verify unicast source reachable-via any
!
ipv6 verify unicast source reachable-via rx
or
ipv6 verify unicast source reachable-via any
!
Reference the Implementing Cisco Express Forwarding section of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information.

Anti-Spoofing ACLs

Manually configured ACLs can provide static anti-spoofing protection against attacks that use known unused and untrusted address space. Commonly, these anti-spoofing ACLs are applied to ingress traffic at the network boundaries as a component of a larger ACL. Anti-spoofing ACLs require regular monitoring because they can frequently change. Spoofing can

be minimized in traffic that originates from the local network by applying outbound ACLs that limit the traffic to valid local addresses.
The following example demonstrates how ACLs can be used to limit IP spoofing. The ACL is applied inbound on the desired interface. The access control entries (ACEs) that make up this ACL are not comprehensive. If administrators configure this type of ACL, they are advised to seek a conclusive, up to date reference.

```
ipv4 access-list ACL-ANTISPOOF-IN
10 deny ipv4 10.0.0.0 0.255.255.255 any
20 deny ipv4 192.168.0.0 0.0.255.255 any
30 permit ipv4 any any
!
interface
ipv4 access-group ACL-ANTISPOOF-IN ingress
!
```

See the IANA IPv4 Address Space Registry for a list of unallocated Internet addresses.
See the Implementing Access Lists and Prefix Lists section of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information about ACL configuration.

Limit CPU Impact of Data Plane Traffic

The primary purpose of routers is to forward packets and frames through the device to final destinations. These packets, which transit the devices deployed throughout the network, can impact the CPU operations of a device. The data plane, which consists of traffic transiting the network device, should be secured to ensure the operation of the management and control planes. If transit traffic can cause a device to process switch traffic, the control plane of a device can be affected, which may lead to an operational disruption.

Features and Traffic Types that Impact the RP and LC CPU

Although not exhaustive, the following list includes the types of data plane traffic that require special CPU processing and that are process-switched by the RP CPU or LC CPU:
ACL Logging: ACL logging traffic consists of any packets that are generated due to a match (permit or deny) of an ACE on which the log keyword is used.
IPv4 Options: Any IPv4 packets with options included must be processed by the CPU. Some may be processed by the LC CPU, while others may require full RP CPU support.
Fragmentation: Any IPv4 packet that requires fragmentation must be passed to the LC CPU for processing.
Time-to-Live (TTL) Expiry: Packets which have a TTL value less than or equal to 1 require ICMP Time Exceeded (ICMP Type 11, Code 0) messages to be sent, resulting in CPU processing. In most cases, processing is done by the LC CPU.
ICMP Unreachable: IPv4 and IPv6 packets that require the generation of ICMP unreachable messages due to routing, MTU, or filtering are handled by either the LC CPU or the RP CPU.
Traffic Requiring an ARP Request: Destinations for which an ARP entry does not exist (CEF glean) require processing by the CPU.
Non-IPv4/IPv6 Traffic: All non-IPv4 and IPv6 traffic is processed by the CPU.
Some functions provided by the Cisco IOS **ip options [ignore | drop]** command and TTL ACL are not directly available in Cisco IOS XR Software; however, most ip options packets are

rate-limited at the microcode level automatically in Cisco IOS XR software. Many packets can be managed by LPTS, which provides a function like the Cisco IOS Software Distributed Mode CoPP (dCoPP) capability. In Cisco IOS XR Software, LPTS is an automatic feature and does not require user configuration. The LPTS feature manages all traffic that would be handled by any CPU (line card of router processor) on a per-line card basis only (aggregate level rate-limiting at the RP level is not supported).
The policing values used within LPTS are not user-configurable until Cisco IOS XR Software Release 3.6. For Cisco IOS XR Software Releases versions 3.6 and later, the LPTS feature policing rates are user configurable.

Traffic Identification and Traceback

At times, administrators must quickly identify and trace back network traffic, especially during an incident response or poor network performance. NetFlow and Classification ACLs are the two primary methods to accomplish traffic identification and traceback when using Cisco IOS XR Software. NetFlow can provide visibility into all traffic on the network. Additionally, NetFlow can be implemented with collectors that can provide long term trending and automated analysis. Classification ACLs are a component of ACLs. Preplanning is required to identify specific traffic and manual intervention during analysis. The following sections provide a brief overview of traffic identification and traceback.

Identify Anomalous Activity by Using NetFlow

NetFlow identifies anomalous and security-related network activity by tracking network flows. NetFlow data can be viewed and analyzed by way of the command line interface (CLI), or NetFlow records can be exported to a commercial or freeware NetFlow collector for aggregation and analysis. NetFlow collectors, through long-term trending, can provide network behavior and usage analysis. NetFlow functions by performing analysis on specific attributes within IP packets and creating flows. NetFlow version 5 is the most commonly used version; however, version 9 is more extensible. NetFlow flow records can be created using sampled traffic data in high-volume environments.
Administrators are advised to consider the following specifications when configuring NetFlow in Cisco IOS XR Software:
Configure a source interface, otherwise, the exporter will remain in a disabled state
Configure a valid record map name for every flow monitor map
Only sampled NetFlow is supported (no static)
Only the IPv4 raw record format is supported
Up to 1 million cache entries are supported per flow monitor
The DSCP value can be set for export packets to avoid policy drops
Export only NetFlow version 9 record format over UDP
Administrators are advised not to use the management interface to export NetFlow records. Exporting NetFlow records using the management interface is not supported on the Cisco IOS XR 12000 Series Router and is not efficient on the Cisco CRS-1 Series routers.
Cisco IOS XR Software Release 3.5/3.6/3.7 supports the NetFlow collection of MPLS packets. It also supports the NetFlow collection of MPLS packets carrying IPv4, IPv6, or both IPv4 and IPv6 payloads. MPLS IPv6 is not supported on the Cisco IOS XR 12000 Series routers.

The sampler map, a subfeature of NetFlow, specifies the rate at which packets (one out of n packets from the flow) are sampled. On high bandwidth interfaces, applying NetFlow processing to every single packet can result in significant CPU utilization. Sampler map configuration is typically used for such high-speed interfaces. If NetFlow is applied in both directions, the flow record packets are policed at the rate of 35,000 packets per second per direction on the Cisco CRS-1, and 25,000 packets per second per direction on the Cisco IOS XR 12000 Series Router. If NetFlow is applied in one direction only, then the flow record packets are policed at the rate of 70,000 packets per second per direction on the Cisco CRS-1, and 50,000 packets per second per direction on the Cisco IOS XR 12000 Series Router. On CRS-1, MSC/B line cards have a higher policer rate than MSC/A.
The following is an example of NetFlow cache summary output from the CLI:
#show flow monitor PE3_monitor cache summary location 0/13/cPU0
Cache summary for Flow Monitor PE3_monitor:
Cache size: 65535
Current entries: 627
High Watermark: 62258
Flows added: 3989508
Flows not added: 0
Ager Polls: 95895

- Active timeout 409
- Inactive timeout 3988471
- TCP FIN flag 1
- Watermark aged 0
- Emergency aged 0
- Counter wrap aged 0
- Total 3988881

Periodic export:

- Counter wrap 0
- TCP FIN flag 0

Flows exported 3988881

See Configuring NetFlow on Cisco IOS XR Software for more information about NetFlow capabilities.

Identify Traffic by Using Classification ACLs

Classification ACLs provide visibility into the traffic that traverses an interface. Classification ACLs do not alter the security policy of a network and are typically constructed to classify individual protocols, source addresses, or destinations. For example, an ACE that permits all traffic could be separated into specific protocols or ports. This more granular classification of traffic into specific ACEs can help provide an understanding of the network traffic because each traffic category has its own hit counter. An administrator may also separate the implicit deny at the end of an ACL into granular ACEs to help identify the types of denied traffic.
An administrator can expedite an incident response by using classification ACLs with the **show access-list** and **clear ip access-list counters EXEC** commands.
The following example illustrates the configuration of a classification ACL to identify fragment traffic prior to a default deny:

ipv4 access-list permit-frag
10 permit tcp any any fragments log
20 permit udp any any fragments
30 permit icmp any any fragments
40 permit ipv4 any any fragments
45 permit icmp any any
50 permit ipv4 any any
To identify the traffic that uses a classification ACL, administrators should use the **show access-list acl-name EXEC** command. The ACL counters can be cleared by using the **clear access-list ipv4 acl-name EXEC** command.
show access-lists permit-frag
ipv4 access-list permit-frag
10 permit tcp any any fragments log **(1386 matches)**
20 permit udp any any fragments
30 permit icmp any any fragments
40 permit ipv4 any any fragments
45 permit icmp any any
50 permit ipv4 any any **(11002 matches)**
!
See the Implementing Access Lists and Prefix Lists section of the Cisco IOS XR IP Addresses and Services Configuration Guide for more information about enabling logging capabilities within ACLs.

MACsec Dynamic Authentication

For additional security during MACsec link bring-up, customers can use the 802.1x based authentication with EAP-TLS. IOS XR platforms support certificate-based EAP-TLS authentication for MACsec link bring-up.

More details can be found in the configuration guide here.

Quantum Safe MACsec

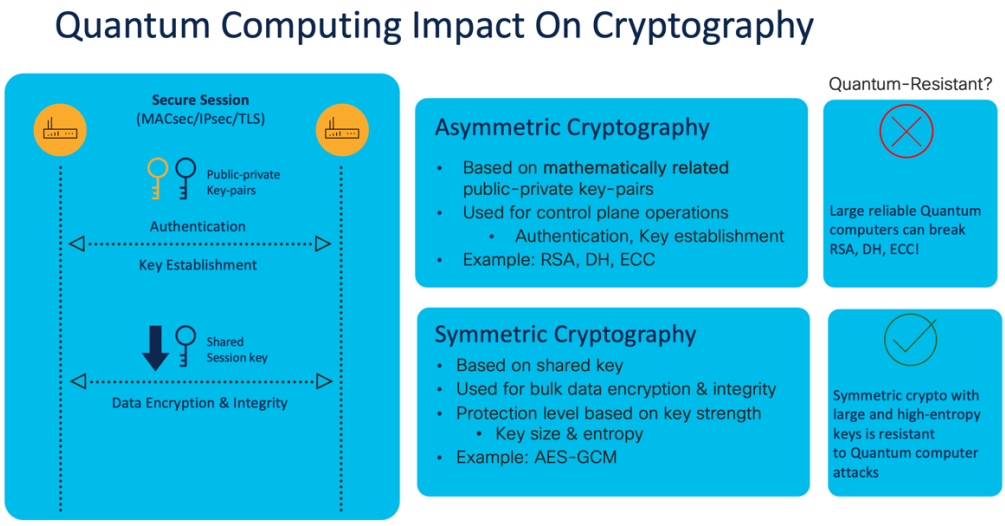
Introduction

Given that there are serious efforts in developing a quantum computer and there are lot of advances too happening in this area, networks operators must start preparing for a post quantum world. The implication of that in security domain is that once there is a sufficiently large and reliable quantum computer available, it can break the current public key algorithms. Even though it might be at least 5 or 10 years away before this could be realistic, the problem is that adversaries can tap the flows today and be able to decrypt the sessions later once quantum computers become real. This is still not an issue for most of the deployment scenarios.

However, if there are sensitive parts of a network that need forward secrecy to be maintained for at least 5 years or more, then the operators must start looking at Quantum-safe security.

Problem Statement

In a typical MACsec session establishment, there are 2 parts (shown in the below diagram). The 1st part is the authentication and key establishment process which is based on asymmetric cryptography and that is where the problem is. Today's asymmetric algos like RSA, Diffe-Hellman, ECC, etc. are all susceptible to large enough quantum computers. The 2nd part which is the actual MACsec session encryption/decryption that is based on symmetric crypto is not susceptible to quantum computers.

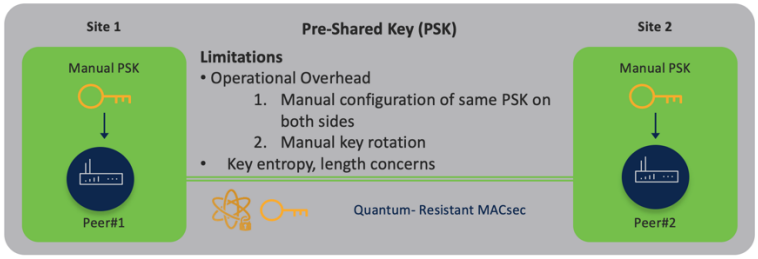


So, in short, any protocols today relying on asymmetric crypto for session establishment are vulnerable.

The long-term option is to wait for post quantum crypto algorithms to be standardized which is still a few years away. So, customers need a short to medium-term option to make their networks Quantum safe.

The 1st option is to use PSK (Pre-Shared Key) method for MACsec link bring-up. This has been supported on all IOS XR platforms from day-1. This doesn't require any additional hardware or software upgrade too.

Pre-Shared Key (PSK) Option



1. MACsec with PSK option is already supported and used by customers.
2. There is no need for additional hardware (like QKD) or software upgrade.
3. Quantum safe as this is based on symmetric cryptography which is Quantum resistant.

The only downside with PSK option is that the key rotation is not automatic and customers who need dynamic authentication for link bring-up cannot use it.

The 2nd option is to use SKS (Session Key Service) or SKIP (Secure Key Integration Protocol) options for MACsec link bring-up.

The SKS service is native to IOS XR and doesn't require any external hardware like Quantum Key Distribution devices. The quantum safe keys can be generated on the routers and the actual keys are never exchanged on the wire. So, even if an eavesdropper is tapping the flows, the keys won't be compromised.

The other option is to use SKIP with MACsec where external Quantum Key Distribution (QKD) hardware can be used to generate the session keys. This makes the keys much stronger as they are generated based on some optical properties. However, today's QKD technology has some limitations on the distances they can support, and this also involves additional cost of procuring and maintaining the QKD hardware.

More details on the Quantum Safe MACsec configuration can be found in the configuration guide here.

The platforms where SKS & SKIP features are supported along with the release information is listed in the below table.

Platform	Release	SKS Support	SKIP Support (With external QKD hardware)
8000	7.4.1	Yes	No
8000	7.9.1	Yes	Yes
NCS55xx	7.9.1	Yes	Yes
NCS57xx	7.9.1	Yes	Yes
NCS540	7.9.1	Yes	Yes
NCS540*	7.10.1	Yes	Yes
ASR9K	7.10.1	Yes	Yes

***Platform Specific Note**
Below are the NCS540 variants that support this feature from 7.10.1 release.

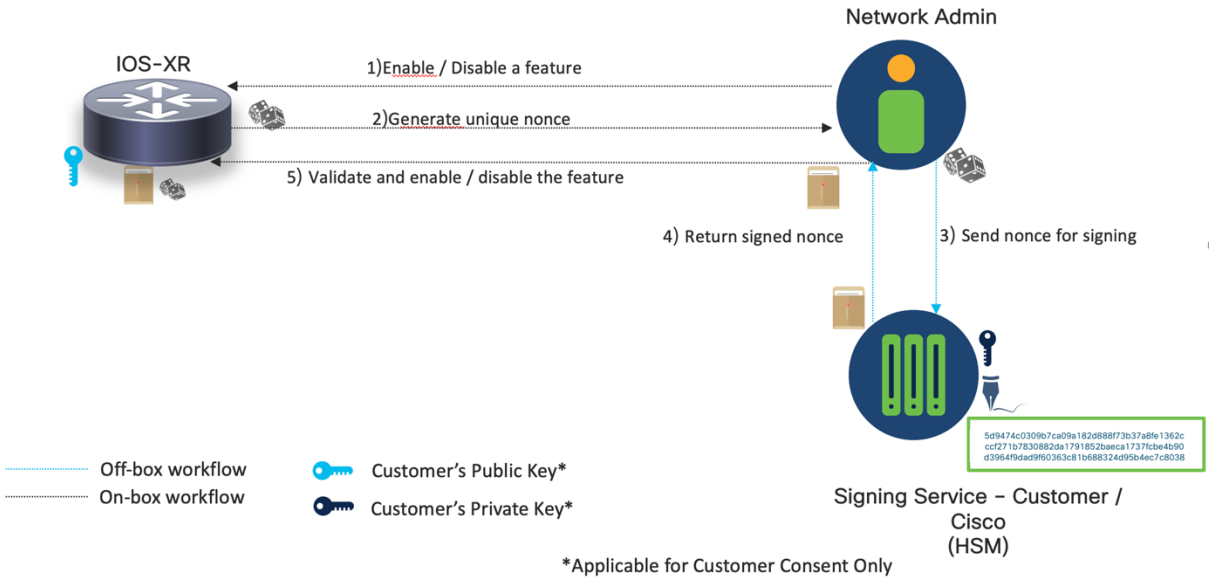
- N540X-16Z4G8Q2C-D/A
- N540X-16Z8Q2C-D
- N540-28Z4C-SYS-D/A
- N540X-12Z16G-SYS-D/A
- N540-12Z20G-SYS-D/A
- N540X-4Z14G2Q-D/A
- N540X-8Z16G-SYS-D/A
- N540X-6Z18G-SYS-D/A
- N540-6Z18G-SYS-D/A
- N540-6Z14S-SYS-D/A
- N540-FH-AGG-SYS
- N540-FH-CSR-SYS

Consent Based Security Features

Introduction

Like 2-factor authentication used for email access or other applications, IOS XR supports additional consent mechanism for some of the sensitive security features. This is essentially to provide additional protection from internal rogue employees and an additional deterrent for external attackers.

As shown in the workflow in the below diagram, when a network operator tries to enable or disable a feature protected by consent mechanism, the router throws an additional challenge to the admin. This challenge must then be signed and returned to the router to proceed further with the intended action.



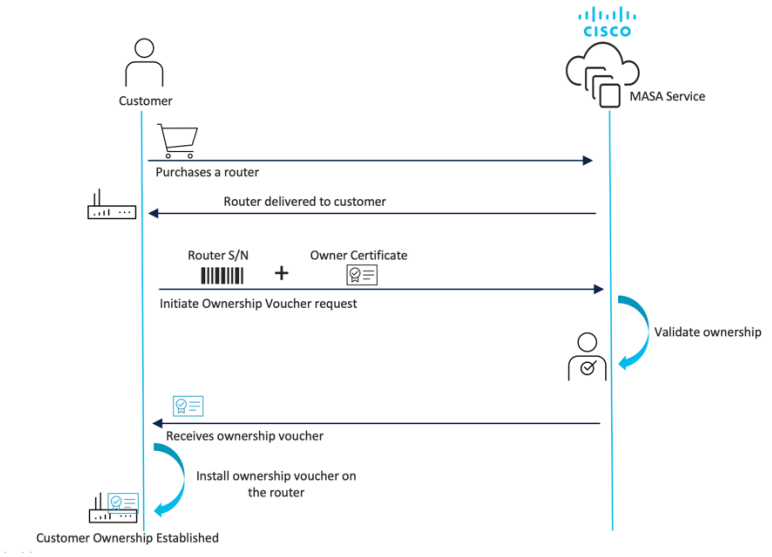
Cisco's IOS XR platforms support 2 types of consent methods as listed below.

1. Cisco Consent Token (CT) where customers don't need any additional setup to sign the challenge thrown by the router. Customers can reach out to Cisco to get the challenge signed and returned to the router.
2. Customers also have the option for setting up the consent mechanism in their own premises. This helps in avoiding the additional step to contact Cisco to get the challenge string signed. To establish this, customer must first establish the ownership of the devices as explained in the next section. The required reference scripts needed for the challenge signing mechanism are made available to customers in GitHub.

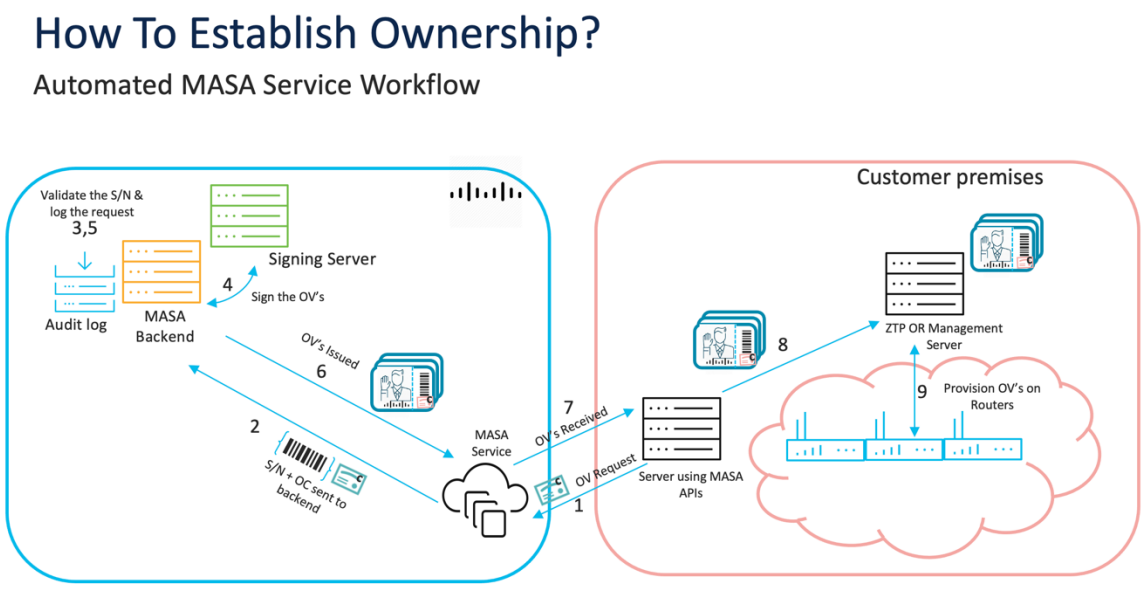
Ownership Establishment

Ownership establishment is the process with which customers can onboard their own Pinned Domain Certificate (PDC) on to the routers they own. Once the trust is established with customer's certificate, it could be used to validate artifacts signed by the customer's own keys. It enables features like customer consent where the challenge thrown by the routers can now be signed by customers own keys.

The concept of ownership is illustrated with the below diagram.



Ownership establishment is done through Ownership Voucher (OV) that is based on RFC8366. Cisco's MASA service could be used by customers to request for an OV. The workflow for establishing ownership is shown in the below diagram.



An OV requests needs serial number of the router on which ownership needs to be established and customer's PDC.

More details on establishing ownership can be found in the feature documentation here.

Gating Lawful Intercepting

Some regulatory agencies mandate the need to have additional consent mechanism to enable Lawful Interception feature. In order to meet such compliance requirements Cisco's IOS XR supports gating lawful interception feature using consent token.

This feature was introduced in IOS XR 7.5.1 release.

This feature is not enabled by default with the base image and requires customers to install the LI-control package (ncs540-lictrl-1.0.0.0-r<*release-number*>.x86_64.rpm).

More details on the feature can be found in the feature documentation here.

Re-Image Protection for Routers

Router theft is a common problem reported by a lot of service providers. Attackers typically steal the cell-site routers deployed in remote locations, re-image them using network (iPXE) or USB boot with a fresh IOS XR image and then re-sell the stolen routers in illegal markets. To deter such thefts, IOS XR now supports re-image protection feature for routers. Once this feature is enabled, malicious users cannot re-image the router which prevents them from re-selling routers in illegal markets. This feature also comes in handy to prevent any malicious attacker from trying to downgrade the image on the router to a known vulnerable version.

Once the feature is enabled, BIOS will check for the feature status flag during the boot process. If BIOS detects that the feature is enabled, both iPXE & USB boot will be disabled. This prevents any unauthorized re-imaging of the routers. The feature flag is stored securely inside the TAm chip so that any adversaries with physical access to the routers are still unable to clear the flag.

The feature is tied with the consent mechanism and as mentioned earlier, either Cisco Consent Token or customer consent mechanism can be used to use the feature.

More details on the feature can be found in the feature documentation here.

Conclusion

This document provides a broad overview of the methods that can be used to secure a Cisco IOS XR system device. Securing the devices increases the overall security of the networks that administrators manage. Administrators can increase the protection of the management, control, and data planes of Cisco IOS XR devices in their network by following the recommendations provided in this document.

Glossary

The following list provides expansions for acronyms and initialisms used in this document:
ARP: Address Resolution Protocol
ACE: Access Control Entries
AAA: Authentication, Authorization, and Accounting
AS: Autonomous System
BGP: Border Gateway Protocol
CLI: Command Line Interface

CPU: Central Processing Unit
Cisco DP: Cisco Discovery Protocol
DF: Data Fragmentation
DNS: Domain Name Service
DoS: Denial of Service
DRP: Distributed Router Processor
dSC: Designated System Controller
eBGP: external Border Gateway Protocol
FIB: Forwarding Information Base
FTP: File Transfer Protocol
FHRPs: First Hop Redundancy Protocols
HSRP: Hot Standby Router Protocol
HTTP: Hypertext Transfer Protocol
ICMP: Internet Control Message Protocol
IGP: Interior Gateway Protocol
IPSec encryption: IP Security encryption
IS-IS: Intermediate System to Intermediate System
Ksh: korn shell
LDP: Label Distribution Protocol
LC: Line Card
LPTS: Local Packet Transport Services
MAC: Media Access Control
MPLS: Multiprotocol Label Switching
MPLS TE: Multiprotocol Label Switching Traffic Engineering
NTP: Network Time Protocol
OPIE: One-time Passwords In Everything
OSPF: Open Shortest Path First
PPP: Point to Point Protocol
PSIRT: Cisco Product Security Incident Response Team
Path MTU: Path Maximum Transmission Unit

Proxy ARP: Proxy Address Resolution Protocol
Qnet: QSSL Network Manager
RFC: Request for Comments
ROMMON: Read Only Memory Monitor
RP: Router Processor
RPL: Routing Policy Language
SC: Shelf Controller
SCP: Secure Copy Protocol
SDR: Secure Domain Router
SFTP: Secure FTP
SSH: Secure Shell Protocol
SNMP: Simple Network Management Protocol
Syslog: System Log
TACACS+: Terminal Access Controller Access-Control System
Telnet: Telecommunication Network
TFTP: Trivial File Transfer Protocol

TTL: Time-to-Live
Unicast RPF: Unicast Reverse Path Forwarding
VRRP: Virtual Router Redundancy Protocol
VRF: Virtual Routing and Forwarding

References

Cisco Security Advisories and Alerts
//tools.cisco.com/security/center/publicationListing.x
Cisco Security Vulnerability Policy
//www.cisco.com/c/en/us/about/security-center/security-vulnerability-policy.html
RFC 2205: Resource ReSerVation Protocol (RSVP)
http://www.ietf.org/rfc/rfc2205.txt
RFC 2385: Protection of BGP Sessions via the TCP MD5 Signature Option
http://www.ietf.org/rfc/rfc2385.txt
RFC 2747: RSVP Cryptographic Authentication
http://www.ietf.org/rfc/rfc2747
RFC 3682: The Generalized TTL Security Mechanism (GTSM)
http://www.ietf.org/rfc/rfc3682.txt
RFC 3882: Configuring BGP to Block Denial-of-Service Attacks
http://www.ietf.org/rfc/rfc3882.txt
Risk Triage for Security Vulnerability Announcements
//www.cisco.com/c/en/us/about/security-center/vulnerability-risk-triage.html
TACACS+ and RADIUS Comparison
//www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml
The BGP TTL Security Hack (BTSH)
http://smakd.potaroo.net/ietf/idref/draft-gill-btsh/index.html

---
