Secure Simple Network Management Protocol

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

Strategies to Secure SNMP

- Avoid Insecure SNMP Community String
- Configure SNMP View
- Configure SNMP Community with Access Control List (ACL)
- Configure SNMPv3
  - Step 1. Identify the Trusted NMS Address
  - Step 2. Create an ACL for the Trusted NMS
  - Step 3. Create an SNMP View
  - Step 4. Create an SNMPv3 Group
  - Step 5. Create the SNMPv3 User
  - Step 6. (Optional) Configure SNMPv3 Traps or Informs
  - Step 7. Disable or Avoid Unnecessary SNMPv1 and SNMPv2c Communities
  - Important Notes
  - Verify
- Configure ACL on Interfaces

How to Check SNMP Errors

Related Information

Introduction

This document describes how to secure your Simple Network Management Protocol (SNMP).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on SNMP features, such as SNMP View and SNMP Version 3 (SNMPv3) that were introduced in Cisco IOS® Software Release. Although some configuration examples and command syntax are based on Cisco IOS®, the core SNMP security concepts also apply to newer Cisco IOS XE platforms. Command syntax, supported authentication and privacy algorithms, platform capabilities, and management plane protection features can vary between Cisco IOS and Cisco IOS XE releases.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Securing SNMP is important because unauthorized SNMP access can expose device information, allow configuration changes when read-write access is enabled, or contribute to denial of service (DoS) conditions.

Earlier SNMP versions, such as SNMPv1 and SNMPv2c, rely on community strings for access control. These community strings function similar to shared passwords, but they are sent in clear text and do not provide strong authentication or encryption. If a community string is reused, exposed in configuration files, or captured from the network, an unauthorized user can be able to poll device information or, in the case of a read-write community, modify the device settings.

Some of the most common SNMPv1/v2c security limitations include:

- Use of well-known community strings such as public or private
- Clear text transmission of community strings
- Lack of per-user authentication
- Lack of encryption for SNMP payloads

For secure deployments, use SNMPv3 with authentication and privacy where supported. SNMPv3 provides user-based authentication and encryption, which helps protect both access control and management data. In addition, you can restrict SNMP access to trusted management stations, avoid unnecessary read-write access, and limit accessible MIB objects with SNMP views.



Note: Validate the exact Command Line Interface (CLI) configuration, feature support, and verification commands for the specific platform and software release before deployment on a production network.

Strategies to Secure SNMP

Avoid Insecure SNMP Community String

Do not use well-known community strings such as public or private. If SNMPv1 or SNMPv2c is required for compatibility, use unique, non-obvious community strings, restrict them with ACLs, and avoid read-write communities unless there is a specific operational requirement.



Note: Prefer SNMPv3 with the authPriv for secure deployments. The authPriv level provides both authentication and encryption. Authentication verifies that SNMP messages come from a valid SNMPv3 user, and privacy encrypts the SNMP payload so management data is not sent in clear text.

Configure SNMP View

An SNMP view limits access to specific MIB subtrees. When a view is applied to a community or SNMPv3 group, only the included MIB objects are accessible. MIB subtrees that are not included are implicitly denied. You can also explicitly exclude specific subtrees from a broader included subtree.

To create or update an SNMP view entry, use the **snmp-server view** global configuration command. To remove the specified SNMP server view entry, use the **no** form of this command.

<#root>

```
snmp-server view <myview-name> <oid-tree> {included | excluded}
```

```
no snmp-server view <myview-name>
```

Syntax Description:

- myview-name — Label for the view record that you update or create. The name is used to reference the record.
- oid-tree — Object Identifier (OID) of the Abstract Syntax Notation One (ASN.1) subtree to be included or excluded from the view. To identify the subtree, specify a text string that consists of numbers, such as 1.3.6.2.4, or a word, such as system. Replace a single sub-identifier with the asterisk (*) wildcard to specify a subtree family; for example 1.3.*.4.
- included or excluded — Type of view. You must specify either included or excluded.

Legacy SNMP documentation, including RFC 1447, introduced the concept of MIB views to control which

portions of the MIB tree are included or excluded from SNMP access. Although RFC 1447 and its party-based SNMPv2 security model are obsolete, the MIB view concept remains relevant in modern SNMP through the View-based Access Control Model (VACM), defined in RFC 3415. Cisco SNMP views use this concept to restrict access to specific MIB subtrees. An SNMP view can be applied to SNMPv1/v2c communities or to SNMPv3 groups.

This example creates a view that includes all objects in the MIB-II system group except for sysServices (System 7) and all objects for interface 1 in the MIB-II interfaces group:

```
<#root>
snmp-server view agon system included
snmp-server view agon system.7 excluded


snmp-server view agon ifEntry.*.1 included
```

In this example, the view allows access to MIB-II and Cisco private MIB objects but excludes the ARP table represented by atEntry:

```
<#root>
snmp-server view myview mib-2 included
snmp-server view myview atEntry excluded
snmp-server view myview cisco included
snmp-server community C1sc03xample view myview RO 11
snmp-server community C1sc03xample2 view myview RW 11
snmp-server contact ciscouser@cisco.com
```

Note: Some of these examples can reference older MIB objects from Cisco IOS releases, such as atTable or atEntry. On newer Cisco IOS XE platforms, equivalent data can use different MIB objects or OIDs.

Note: Before deploying an SNMP view, verify that the MIB object name or OID exists on the target platform and software release. Use SNMP Object Navigator for additional information.

Configure SNMP Community with Access Control List (ACL)

If SNMPv1 or SNMPv2c is required, restrict each community string to trusted Network Management System (NMS) addresses with an ACL. The ACL associated with the SNMP community controls which source IP addresses can use that community string to send SNMP requests to the device.

SNMP communication commonly includes two different traffic types:

- Polling / requests: The NMS sends SNMP GET, GETNEXT, GETBULK, or SET requests to the network device. These requests usually target UDP port 161 on the managed device.
- Traps / notifications: The network device sends unsolicited SNMP trap or inform messages to the NMS to report events, alarms, or state changes. These notifications usually target UDP port 162 on the NMS.

Use different community strings for polling requests and trap notifications. This separation limits exposure if one community string is discovered. For example, if an attacker learns the trap community string from a notification packet, that string must not also allow SNMP polling access to the device.

<#root>

access-list 10 permit <NMS_IP>

<#root>

snmp-server community <RO_POLLING_COMMUNITY> RO 10

<#root>

snmp-server host <NMS_IP> version 2c <TRAP_NOTIFICATION_COMMUNITY>

In this example:

- <RO_POLLING_COMMUNITY> is used by the NMS to poll the device.
- <TRAP_NOTIFICATION_COMMUNITY> is used by the device when it sends traps or notifications to the NMS.
- ACL 10 restricts polling access so only <NMS_IP> can use the polling community string.
- The trap notification community string is not reused for SNMP polling access.

If a trap community string is configured, verify that it does not unintentionally permit SNMP polling access. On some Cisco IOS software releases, a community string used for traps can also become usable for SNMP access unless explicitly restricted or disabled.

To prevent polling access with the trap community string, explicitly deny its use as an SNMP community:

<#root>

```
access-list 99 deny any
```

<#root>

snmp-server community <TRAP_NOTIFICATION_COMMUNITY> RO 99

Avoid configuring read-write communities unless there is a specific operational requirement. If read-write access is required, restrict it to a dedicated management station, use a separate community string, and apply a restrictive ACL. Prefer SNMPv3 with authentication and privacy for secure deployments.

Configure SNMPv3

Use SNMPv3 with authentication and privacy for secure SNMP deployments. SNMPv3 provides user-based authentication and encryption, unlike SNMPv1 and SNMPv2c, which rely on community strings.

For secure deployments, use the SNMPv3 **authPriv** security level:

- Authentication verifies that the SNMP message comes from a valid SNMPv3 user.
- Privacy encrypts the SNMP payload so management data is not sent in clear text.
- Avoid noAuthNoPriv and authNoPriv for secure production deployments.
- Avoid weak or legacy algorithms such as MD5 and DES when stronger options are supported by the platform.



Note: Algorithm support varies by Cisco platform and software release. Use the strongest authentication and privacy algorithms supported by the device, such as SHA or SHA-2 for authentication and AES for privacy.

SNMPv3 Configuration Steps

Step 1. Identify the Trusted NMS Address

Decide which Network Management System is allowed to poll the device.

Example value:

- NMS IP address: <NMS_IP>
- Device management IP address: <DEVICE_MGMT_IP>

Note: Only trusted management stations must be allowed to access SNMP services.

Step 2. Create an ACL for the Trusted NMS

Create an **ACL** that permits SNMP access only from the trusted NMS.

Example:

<#root>

access-list 10 permit <NMS_IP> access-list 10 deny any log

This ACL limits which source IP addresses can use the SNMPv3 group. The deny any log entry helps identify unauthorized SNMP access attempts in syslog.

Step 3. Create an SNMP View

Create an **SNMP view** to limit which MIB objects the SNMPv3 user can access.

Example:

<#root>

snmp-server view <VIEW_NAME> mib-2 included

<#root>

snmp-server view <VIEW_NAME> cisco included

This example allows access to the MIB-II subtree and Cisco private MIB objects. If a more restrictive view is required, explicitly exclude sensitive or unnecessary MIB subtrees.

Example:

```
<#root>

snmp-server view <VIEW_NAME> <MIB_FAMILY> excluded
```



Note: An SNMP view reduces exposure by allowing the NMS to access only the MIB objects required for monitoring.

Step 4. Create an SNMPv3 Group

Create an **SNMPv3 group** that uses the priv security level and references the SNMP view and ACL.

Example:

```
<#root>

snmp-server group <GROUP_NAME> v3 priv read <VIEW_NAME> access 10
```

This command does three important things:

- v3 specifies SNMPv3.
- priv requires both authentication and encryption.
- read <VIEW_NAME> limits the readable MIB objects with the SNMP View previously defined.
- access 10 restricts SNMP access to the trusted NMS source IP address with the ACL defined.



Note: For most monitoring use cases, read-only access is usually sufficient. Configure write access only when there is a specific operational requirement.

Step 5. Create the SNMPv3 User

Create an **SNMPv3 user** and assign it to the SNMPv3 group.

Example:

<#root>

snmp-server user <USERNAME> <GROUP_NAME> v3 auth sha <AUTH_PASSWORD> priv aes 256 <PRIV_PASSWORD>

This command creates a user with:

- SNMPv3 authentication
- SHA authentication, where supported
- AES with 256 bit encryption, where supported
- Membership in the restricted SNMPv3 group



Note: Use strong, unique passwords for authentication and privacy. Do not reuse the same password for both values.



Note: If newer platforms support stronger options, such as SHA-2 or stronger AES variants, use the strongest option supported by the device and NMS.

Step 6. (Optional) Configure SNMPv3 Traps or Informs

If the device needs to send notifications to the NMS, configure the **SNMP host** with SNMPv3.

Example:

<#root>

snmp-server host <NMS_IP> version 3 priv <USERNAME>

Traps and informs are notifications sent from the device to the NMS. They are different from polling requests, where the NMS sends SNMP queries to the device. Use SNMPv3 notifications so trap or inform traffic also benefits from authentication and privacy.

If needed, also configure the trap source interface:

<#root>

snmp-server trap-source <INTERFACE>



Note: Using a consistent trap source helps the NMS identify the device and allows firewalls or ACLs to match predictable source addresses.

Step 7. Disable or Avoid Unnecessary SNMPv1 and SNMPv2c Communities

If SNMPv1 or SNMPv2c is not required, do not configure community strings or remove unused community strings.

Example:

<#root>

no snmp-server community <COMMUNITY_STRING>



Note: This reduces the risk of unauthorized SNMP access through clear-text community strings.

The next configuration snippet shows an example that permits SNMPv3 polling only from NMS 10.189.5.161, limits readable MIB objects with an SNMP view, and requires authentication and encryption.

<#root>

Router#configure terminal
Router(config)#

access-list 10 permit 10.189.5.161

Router(config)#

access-list 10 deny any log

Router(config)#!
Router(config)#

snmp-server view ciscoview mib-2 included

Router(config)#

snmp-server view ciscoview cisco included

```
Router(config)#
```

```
snmp-server view ciscoview ipNetToMediaEntry.2 excluded
```

```
Router(config)#
```

```
snmp-server group myciscogroup v3 priv read ciscoview access 10
```

```
Router(config)#
```

```
snmp-server user ciscouser myciscogroup v3 auth sha C1sc03xampl3 priv aes 256 C1sc03xampl32
```

Important Notes

- Configure the SNMP view before assigning it to the SNMPv3 group.
- If a local SNMP engine ID is manually configured, configure it before creating SNMPv3 users. Changing the engine ID after user creation can require the SNMPv3 users to be recreated.
- SNMPv3 user information does not appear fully in **show running-config** for security reasons. Use **show snmp user** for verification.
- Exact MIB object syntax and supported algorithms vary by Cisco IOS and Cisco IOS XE release. Validate the command syntax for the target platform before deployment.

Verify

Use these commands to verify the local SNMP configuration:

show snmp user
show snmp group
show snmp view
show snmp
show ip access-lists
show running-config | section snmp

<#root>

ISP-C#

```
show snmp user
```

```
User name: ciscouser
Engine ID: 80000009030000A38EB6F800
storage-type: nonvolatile        active
Authentication Protocol:
```

```
SHA
```

Privacy Protocol:

AES256

Group-name: myciscogroup

```
ISP-C#show snmp group
groupname: ILMI                                 security model:v1
contextname: <no context specified>             storage-type: permanent
readview : *ilmi                                writeview: *ilmi
notifyview: <no notifyview specified>
row status: active

groupname: ILMI                                 security model:v2c
contextname: <no context specified>             storage-type: permanent
readview : *ilmi                                writeview: *ilmi
notifyview: <no notifyview specified>
row status: active

groupname: myciscogroup                         security model:v3 priv
contextname: <no context specified>             storage-type: nonvolatile
readview : ciscoview                            writeview: <no writeview specified>
notifyview: <no notifyview specified>
row status: active      access-list: 10

ISP-C#
```

show snmp view

```
*ilmi system - included permanent active
*ilmi atmForumUni - included permanent active
cac_view pimMIB - included read-only active
cac_view msdpMIB - included read-only active
cac_view system - included read-only active
cac_view interfaces - included read-only active
cac_view ip - included read-only active
cac_view ospf - included read-only active
cac_view bgp - included read-only active
cac_view ifMIB - included read-only active
cac_view nhrpMIB - included read-only active
cac_view ipMRouteStdMIB - included read-only active
cac_view igmpStdMIB - included read-only active
cac_view pimStdMIB - included read-only active
cac_view mgmdStdMIB - included read-only active
cac_view ospfv3MIB - included read-only active
cac_view ipForward - included read-only active
cac_view ipTrafficStats - included read-only active
cac_view ospfTrap - included read-only active
cac_view sysUpTime.0 - included read-only active
cac_view ciscoPingMIB - included read-only active
cac_view ciscoPimMIB - included read-only active
cac_view ciscoBgp4MIB - included read-only active
cac_view ciscoIfExtensionMIB - included read-only active
cac_view ciscoEigrpMIB - included read-only active
cac_view ciscoCefMIB - included read-only active
cac_view ciscoNhrpExtMIB - included read-only active
cac_view ciscoIpMRouteMIB - included read-only active
cac_view ciscoDlcSwitchMIB - included read-only active
cac_view ciscoExperiment.101 - included read-only active
```

```
cac_view ciscoIetfIsisMIB - included read-only active
cac_view ifIndex - included read-only active
cac_view ifDescr - included read-only active
cac_view ifType - included read-only active
cac_view ifAdminStatus - included read-only active
cac_view ifOperStatus - included read-only active
cac_view snmpTraps.3 - included read-only active
cac_view snmpTraps.4 - included read-only active
cac_view snmpTrapOID.0 - included read-only active
cac_view snmpMIB.1.4.3.0 - included read-only active
cac_view lifEntry.20 - included read-only active
cac_view cciDescriptionEntry.1 - included read-only active
ciscoview mib-2 - included nonvolatile active
ciscoview cisco - included nonvolatile active
ciscoview ipNetToMediaEntry.2 - excluded nonvolatile active
v1default iso - included permanent active
v1default internet - included permanent active
v1default snmpUsmMIB - excluded permanent active
v1default snmpVacmMIB - excluded permanent active
v1default snmpCommunityMIB - excluded permanent active

ISP-C#
```

```
show ip access-lists
```

```
Standard IP access list 10
    10 permit 10.189.5.161 (26 matches)
    20 deny   any log (6 matches)
Extended IP access list meraki-fqdn-dns

ISP-C#
```

```
show running-config | section snmp
```

```
snmp-server group myciscogroup v3 priv read ciscoview access 10
snmp-server view ciscoview mib-2 included
snmp-server view ciscoview cisco included
snmp-server view ciscoview ipNetToMediaEntry.2 excluded
```

You can also do an SNMP walk for the MIB-II System tree:

<#root>

ciscouser@Desktop %

```
snmpwalk -v3 -l authPriv -u ciscouser -a SHA -A 'C1sc03xampl3' -x AES -X 'C1sc03xampl32' 10.207.214.25 s
```

```
SNMPv2-MIB::sysDescr.0 = STRING: Cisco IOS Software [Cupertino], ASR1000 Software (X86_64_LINUX_IOSD-UN
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2023 by Cisco Systems, Inc.
Compiled Wed 26-Jul-23 10:29 b
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.2348
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (107487239) 12 days, 10:34:32.39
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: ISP-C
```

```
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 78
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

The next command and output shows that you cannot get the ipNetToMediaEntry.2 group:

<#root>

ciscouser@Desktop %

snmpwalk -v3 -l authPriv -u ciscouser -a SHA -A 'C1sc03xampl3' -x AES -X 'C1sc03xampl32' 10.207.214.25 i

IP-MIB::ipNetToMediaPhysAddress = No Such Object available on this agent at this OID

The next command and output result is for the Cisco Local System group:

<#root>

ciscouser@Desktop %

snmpwalk -v3 -l authPriv -u ciscouser -a SHA -A 'C1sc03xampl3' -x AES -X 'C1sc03xampl32' 10.207.214.25 .

```
SNMPv2-SMI::enterprises.9.2.1.1.0 = STRING: "
17.15(4r)
"
SNMPv2-SMI::enterprises.9.2.1.2.0 = STRING: "LocalSoft"
SNMPv2-SMI::enterprises.9.2.1.3.0 = STRING: "ISP-C"
SNMPv2-SMI::enterprises.9.2.1.4.0 = ""
```

SNMPv2-SMI::enterprises.9.2.1.5.0 = IpAddress: 10.189.5.161

```
SNMPv2-SMI::enterprises.9.2.1.6.0 = IpAddress: 0.0.0.0
SNMPv2-SMI::enterprises.9.2.1.8.0 = INTEGER: -796648892
SNMPv2-SMI::enterprises.9.2.1.9.0 = INTEGER: 1646
SNMPv2-SMI::enterprises.9.2.1.10.0 = INTEGER: 1000
SNMPv2-SMI::enterprises.9.2.1.11.0 = INTEGER: 1557645
SNMPv2-SMI::enterprises.9.2.1.12.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.13.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.14.0 = INTEGER: 104
SNMPv2-SMI::enterprises.9.2.1.15.0 = INTEGER: 1200
SNMPv2-SMI::enterprises.9.2.1.16.0 = INTEGER: 1200
SNMPv2-SMI::enterprises.9.2.1.17.0 = INTEGER: 2500
SNMPv2-SMI::enterprises.9.2.1.18.0 = INTEGER: 2684018
SNMPv2-SMI::enterprises.9.2.1.19.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.20.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.21.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.22.0 = INTEGER: 600
SNMPv2-SMI::enterprises.9.2.1.23.0 = INTEGER: 900
SNMPv2-SMI::enterprises.9.2.1.24.0 = INTEGER: 900
SNMPv2-SMI::enterprises.9.2.1.25.0 = INTEGER: 2000
SNMPv2-SMI::enterprises.9.2.1.26.0 = INTEGER: 26757370
SNMPv2-SMI::enterprises.9.2.1.27.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.28.0 = INTEGER: 0
```

```
SNMPv2-SMI::enterprises.9.2.1.29.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.30.0 = INTEGER: 1536
SNMPv2-SMI::enterprises.9.2.1.31.0 = INTEGER: 900
SNMPv2-SMI::enterprises.9.2.1.32.0 = INTEGER: 899
SNMPv2-SMI::enterprises.9.2.1.33.0 = INTEGER: 1800
SNMPv2-SMI::enterprises.9.2.1.34.0 = INTEGER: 8891131
SNMPv2-SMI::enterprises.9.2.1.35.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.36.0 = INTEGER: 1
SNMPv2-SMI::enterprises.9.2.1.37.0 = INTEGER: 1
SNMPv2-SMI::enterprises.9.2.1.38.0 = INTEGER: 5024
SNMPv2-SMI::enterprises.9.2.1.39.0 = INTEGER: 100
SNMPv2-SMI::enterprises.9.2.1.40.0 = INTEGER: 100
SNMPv2-SMI::enterprises.9.2.1.41.0 = INTEGER: 300
SNMPv2-SMI::enterprises.9.2.1.42.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.43.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.44.0 = INTEGER: 1
SNMPv2-SMI::enterprises.9.2.1.45.0 = INTEGER: 1
SNMPv2-SMI::enterprises.9.2.1.46.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.47.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.48.0 = IpAddress: 0.0.0.0
SNMPv2-SMI::enterprises.9.2.1.49.0 = ""
SNMPv2-SMI::enterprises.9.2.1.51.0 = IpAddress: 0.0.0.0
SNMPv2-SMI::enterprises.9.2.1.52.0 = ""
SNMPv2-SMI::enterprises.9.2.1.56.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.57.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.58.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.59.0 = INTEGER: 4972
SNMPv2-SMI::enterprises.9.2.1.60.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.61.0 = STRING: "cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA  95134-1706
U.S.A.
Ph +1-408-526-4000
Customer service 1-800-553-6387 or +1-408-526-7208
24HR Emergency 1-800-553-2447 or +1-408-526-7209
Email Address tac@cisco.com
World Wide Web http://www.cisco.com"
SNMPv2-SMI::enterprises.9.2.1.62.0 = INTEGER: 18024
SNMPv2-SMI::enterprises.9.2.1.63.0 = INTEGER: 20
SNMPv2-SMI::enterprises.9.2.1.64.0 = INTEGER: 20
SNMPv2-SMI::enterprises.9.2.1.65.0 = INTEGER: 33
SNMPv2-SMI::enterprises.9.2.1.66.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.67.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.68.0 = INTEGER: 1
SNMPv2-SMI::enterprises.9.2.1.69.0 = INTEGER: 1
SNMPv2-SMI::enterprises.9.2.1.70.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.71.0 = INTEGER: 0
SNMPv2-SMI::enterprises.9.2.1.72.0 = IpAddress: 0.0.0.0
SNMPv2-SMI::enterprises.9.2.1.73.0 = STRING: "bootflash:asr1000-universalk9.17.09.04.SPA.bin"
SNMPv2-SMI::enterprises.9.2.1.74.0 = INTEGER: 3
```

Configure ACL on Interfaces

Interface ACLs can be useful as an additional protection layer to limit which hosts can send SNMP traffic to a device interface. They are especially helpful on platforms that do not support more targeted control-plane protection features, such as Control Plane Policing (CoPP), or management-plane ACLs.

An interface ACL can block unauthorized SNMP packets before they are processed by the device, reducing exposure to unwanted polling, brute-force community string attempts, or excessive SNMP traffic from untrusted sources.



Caution: Be careful when applying interface ACLs. An ACL applied with **ip access-group** filters traffic in the selected direction and can affect both traffic destined to the router and transit traffic. Always include all required permit statements and validate the ACL in a maintenance window or lab before production deployment.

<#root>

ip access-list extended SNMP-MGMT-IN permit udp host <NMS_IP> host <DEVICE_MGMT_IP> eq snmp deny udp any host <DEVICE_MGMT_IP> e

The ACL must then be applied to all interfaces with these configuration commands:

<#root>

interface <INTERFACE> ip access-group SNMP-MGMT-IN in

In this example:

- SNMP polling from the trusted NMS to the device management IP address is permitted.
- SNMP polling from other sources to the device management IP address is denied and logged.
- Other IP traffic is permitted by the final permit ip any any statement.



Note: You can adapt this example to your network design. If the platform supports CoPP, iACLs, or management-plane ACLs, those options can provide more targeted control-plane protection. For devices that have many IP addresses configured, or many hosts that need to communicate with the router, this is not always a scalable solution. The Protecting Your Core: Infrastructure Protection Access Control Lists document presents guidelines and recommended deployment techniques for ACLs.

How to Check SNMP Errors

Configure the **SNMP community ACLs** with the **log** keyword. Monitor syslog for failed attempts, as shown.

<#root>

access-list 10 deny any log
snmp-server community <RO_POLLING_COMMUNITY> **RO 10**

When someone tries to access the router with the community public, you see a syslog similar to this:

<#root>

%SEC-6-IPACCESSLOGS: list 10 denied 172.16.1.15 packet

This output means that access-list 10 has denied five SNMP packets from the host 172.16.1.15.

Periodically check SNMP for errors with the **show snmp** command, as shown here:

<#root>

ISP-C#

show snmp

```
Chassis: TKM21110169
186 SNMP packets input
 0 Bad SNMP version errors 58 Unknown community name
    0 Illegal operation for community name supplied
 0 Encoding errors
    115 Number of requested variables
    0 Number of altered variables
    2 Get-request PDUs
    113 Get-next PDUs
    0 Set-request PDUs
    0 Input queue packet drops (Maximum queue size 1000)
126 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    0 Trap PDUs
Packets currently in SNMP process input queue: 0
SNMP global trap: disabled

SNMP logging: disabled
```

Watch the highlighted counters for unexpected increases in error rates that can indicate attempted exploitation of these vulnerabilities. To report any security issue, refer to Cisco Product Security Incident Response.

Related Information

- Cisco Security Advisories SNMP Vulnerabilities
- Cisco Technical Support & Downloads