CCNA Exam Prep: Back to Networking Basics with Hank Preston and Patrick Gargano -- Season 2

Simplifying SNMP: Delivering Data for Network Operation Dashboards

Network Monitoring is critical to security and troubleshooting tasks. As your network grows and evolves, centralized monitoring becomes even more important. SNMP is a protocol that allows you to remotely monitor a wide range of settings and counters, be alerted when there are changes, and even remotely make configuration changes.

In this lab you will explore how SNMP works and how to configure and verify SNMP on Cisco devices.

Related CCNA v1.1 exam topic:

- 4.4 Explain the function of SNMP in network operations

In this lab, we will:

- Explore SNMP and its different versions
- Configure and verify SNMPv2c
- Configure and verify SNMPv3
- Explore the SNMP message format and MIB using the snmpwalk command

Setup and Scenario

In this set of lab-based demonstrations, you are a network engineer tasked with configuring SNMPv2c on a Cisco switch and SNMPv3 on a Cisco router.

You've been asked to:

- Configure the sw01 switch to send SNMPv2c traps to the NMS server.
- Configure the rtr01 router to send SNMPv3 traps to the NMS server.

*Be sure to **START** the lab before continuing to the demo lab.*

> Note: The credentials for all devices are shown in the CML topology.

Part 1: Explore SNMP and its different versions

Before we jump into configuring SNMP, let's explore how SNMP operates and how it evolved over time.

SNMP has become the standard for network management. It is a simple, easy-to-implement protocol and is supported by nearly all vendors. SNMP defines how management information is exchanged between SNMP managers and SNMP agents. It uses the UDP transport mechanism on ports 162 and 162 to retrieve and send management information, such as MIB variables.

SNMP is typically used to gather environment and performance data such as device CPU usage, memory usage, interface traffic, interface error rate, and so on.

The SNMP manager periodically polls the SNMP agents on managed devices by querying the device for data. Agents can generate SNMP traps, which are unsolicited notifications that are sent from agent to manager. SNMP traps are event-based and provide almost real-time event notifications.

There are three important concepts when it comes to SNMP:

- The SNMP manager or network management system (NMS) collects management data from managed devices via polling or trap messages.
- The SNMP agent is found on a managed network device, it locally organizes data and sends it to the manager.
- The Management Information Base (MIB) represents a virtual information storage location that contains collections of managed objects. Within the MIB, there are objects that relate to different defined MIB modules (for example, the interface module). Objects in the MIB are referenced by their object ID (OID), which specifies the path from the tree root to the object. For example, system identification data is located under 1.3.6.1.2.1.1. Some examples of system data include the system name (OID 1.3.6.1.2.1.1.5), system location (OID 1.3.6.1.2.1.1.6), and system uptime (OID 1.3.6.1.2.1.1.3).

You can use the Cisco SNMP SNMP Object Navigator to locate and identify different OID values

There are three different versions of SNMP:

- SNMP version 1: SNMPv1 is the initial version of SNMP. SNMPv1 security is based on communities that are nothing more than passwords: plaintext strings that allow any SNMP-based application that knows the strings to gain access to the management information of a device. There are typically three communities in SNMPv1: read-only, read-write, and trap.
- SNMP version 2c: SNMPv2 was the first attempt to fix SNMPv1 security flaws. However, SNMPv2 never really took off. The only prevalent version of SNMPv2 today is SNMPv2c, which contains SNMPv2 protocol enhancements but leaves out the security features that no one could agree on. The letter "c" designates v2c as being "community-based," which means that it uses the same authentication mechanism as v1: community strings.
- SNMP version 3: SNMPv3 is the latest version. It adds support for strong authentication and private communication between managed entities. You can define a secure policy for each group, and optionally you can limit the IP addresses to which its members can belong. You have to define encryption and hashing algorithms and passwords for each user. SNMPv3 introduces three levels of security:
    - Security level `noAuthNoPriv`: No authentication is required, and no privacy (encryption) is provided.
    - Security level `authNoPriv`: Authentication is required, but no encryption is provided.
    - Security level `authPriv`: In addition to authentication, encryption is also used.

Part 2: Configure and verify SNMPv2c on sw01

Let's deploy SNMPv2c on the sw01 switch to send traps to the SNMP_NMS device.

Step 1

Configure an access-list for SNMPv2 use on sw01.

```
sw01(config)# ip access-list standard SNMP-NMS
sw01(config-std-nacl)# permit host 192.168.100.100
sw01(config-std-nacl)# exit
```

This ACL will be used to specify exactly where SNMP **get** and **set** messages should be coming from. In this lab, the 192.168.100.0/24 network is the management network, and the SNMP manager is located at 192.168.100.100

Note that the Alpine Linux server in this lab is not yet configured as a NMS.

Step 2

Configure general SNMP information. Specify a location for the device, some contact information, and a chassis value.

```
sw01(config)# snmp-server location RCD
sw01(config)# snmp-server contact ccnaprep@example.com
sw01(config)# snmp-server chassis-id Cisco IOL-L2 Switch sw01
```

Step 3

SNMPv2c using a community string-based authentication. Access can be limited further by using an access list. Create a read-only community named CCNAPREP that is limited by the SNMP-NMS ACL

```
sw01(config)# snmp-server community CCNAPREP ro SNMP-NMS
```

Step 4

Configure 192.168.100.100 as a trap receiver using SNMPv2c and the community CCNAPREP.

```
sw01(config)# snmp-server host 192.168.100.100 version 2c CCNAPREP
```

Step 5

Configure interface index persistence.

```
sw01(config)# snmp-server ifindex persist
```

Network monitoring systems record throughput and other interface statistics using SNMP polling. Each interface is referenced by its unique index number, which is dynamically assigned by the IOS during bootup. The index of each interface can be determined with the command `show snmp mib ifmib ifindex`. The dynamic assignment aspect of this can be problematic for documentation. Therefore, it is a good idea to instruct the system to keep a persistent list of interfaces, rather than a dynamic one. The use of this command creates a file stored in NVRAM.

Use the `show snmp mib ifmib ifindex` command to verify the current interface indexes.

```
sw01# show snmp mib ifmib ifindex
Ethernet0/1: Ifindex = 2
Ethernet0/3: Ifindex = 4
Loopback0: Ifindex = 8
Ethernet0/0: Ifindex = 1
Null0: Ifindex = 6
Vlan10: Ifindex = 7
Ethernet0/2: Ifindex = 3
SR0: Ifindex = 5
```

Notice that the Loopback 0 interface in the output above has an index value of 8. Your actual values might differ.

Step 6

Enable SNMP to send a trap to the NMS server when an interface on the switch goes down or up.

```
sw01(config)# snmp-server enable traps snmp linkdown linkup
```

This final SNMP configuration command actually enables the forwarding of traps to the configured trap receiver. As a part of this command, traps can be limited (as they can be in the `snmp-server host` command). For this lab, you will send a trap to the NMS anytime an interface goes down or up.

Step 7

Start a Packet Capture on the link between rtr01 and the NMS server and filter the capture to UDP port 162.

Step 8

Verify SNMP configuration

To verify that traps are being sent, issue the command `debug snmp packets` and then shutdown the Loopback 0 interface. You should see debug output indicating that an SNMP packet was sent. It might take a few moments for the switch to start sending traps and for them to appear in the packet capture.

```
sw01# debug snmp packets
sw01# config t
```

```
sw01(config)# interface loopback 0
sw01(config-if)# shut
sw01(config-if)#
*Apr 26 01:10:18.534: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to down
*Apr 26 01:10:18.534: %LINK-5-CHANGED: Interface Loopback0, changed state to administratively down
sw01(config-if)#
*Apr 26 01:10:18.534: SNMP: Queuing packet to 192.168.100.100
*Apr 26 01:10:18.534: SNMP: V2 Trap, reqid 10, errstat 0, erridx 0
 sysUpTime.0 = 2047407
 snmpTrapOID.0 = snmpTraps.3
 ifIndex.8 = 8
 ifDescr.8 = Loopback0
 ifType.8 = 24
 lifEntry.20.8 = administratively down
*Apr 26 01:10:18.784: SNMP: Packet sent via UDP to 192.168.100.100
```

Notice in the output that a packet was sent to 192.168.100.100 using UDP, and that the message references the Loopback0 interface and shows that the interface was manually shutdown.

Step 9

Verify the SNMP packet in the Packet Capture.

Click the captured packet and open the SNMP line in the packet details. Expand all the available lines. You will see that CML does not interpret or convert the ASCII values for the individual OID entries in the SNMP packet. You will need Wireshark to help you with that. Wireshark is a free and open-source network protocol analyzer used for troubleshooting, analyzing, and debugging network problems. It captures and displays network traffic, allowing users to examine data at the packet level. This tool is valuable for network administrators, security professionals, and developers to understand how applications communicate and identify potential security issues. Although CML's Packet Capture tool is very powerful, for advanced packet analysis, Wireshark is invaluable.

> You can download Wireshark here.

In CML, click the `Download` button in the Packet Capture window to save the PCAP to your PC. Open the PCAP file using Wireshark. Expand the SNMP line and all its entries. This time you will be able to read in clear text the information contained in the SNMP packet. You should see the index value of the interface (8), the name of the interface (Loopback0), and the status of the interface (administratively down).

Step 10

Verify general SNMP settings using the `show snmp` command.

```
sw01# show snmp
Chassis: Cisco IOL-L2 Switch sw01
Contact: ccnaprep@example.com
Location: RCD
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Set-request PDUs
    0 Input queue packet drops (Maximum queue size 1000)
    0 Dispatcher queue packet drops (Maximum queue size 75)
1 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
    1 Trap PDUs
Packets currently in SNMP process input queue: 0, max 1000
Packets currently in SNMP PDU dispatcher queue: 0, max 75
SNMP global trap: enabled

SNMP logging: enabled
    Logging to 192.168.100.100.162, 0/10, 1 sent, 0 dropped.
```

The output confirms the basic SNMP settings and how many trap messages were sent to the NMS.

Step 11

Verify the SNMP configuration found in the running-config on sw01.

```
sw01# show run | section snmp
snmp-server community CCNAPREP RO SNMP-NMS
snmp-server location RCD
snmp-server contact ccnaprep@example.com
snmp-server chassis-id Cisco IOL-L2 Switch sw01
snmp-server enable traps snmp linkdown linkup
snmp-server host 192.168.100.100 version 2c CCNAPREP
snmp ifmib ifindex persist
```

Part 3: Configure and verify SNMPv3 on rtr01

Let's deploy SNMPv3 on the rtr01 router to send traps to the SNMP-NMS device. The configuration for SNMPv3 is quite different to what you did on sw01 for SNMPv2c. For SNMPv3, instead of a community string, you need to configure the following:

- SNMP view: Views specify which MIB objects are visible to a particular group.
- SNMP group: Groups are used to combine users and assign them to specific views and security levels.
- SNMP user: Once groups and views are defined, users are assigned to a specific group.

In summary, views define what information is visible, groups define who can access that information, and users are assigned to groups to inherit the access permissions. This layered approach provides flexibility and granular control over SNMPv3 access on Cisco devices.

Step 1

Configure an access-list for SNMPv3 use on rtr01.

```
rtr01(config)# ip access-list standard SNMP-NMS
rtr01(config-std-nacl)# permit host 192.168.100.100
rtr01(config-std-nacl)# exit
```

This ACL is identical to the ACL you configured on sw01 and limits access to the router's MIB .

Step 2

Configure a SNMPv3 view called SNMP-RO that includes the entire `iso` tree from the MIB.

```
rtr01(config)# snmp-server view SNMP-RO iso included
```

In a production network, you would want to restrict the view to a smaller subset of MIBs.

Step 3

Configure a read-only SNMPv3 group with the highest level of security supported and assigned the ACL defined in Step 1.

```
rtr01(config)# snmp-server group ADMIN v3 priv read SNMP-RO access SNMP-NMS
```

A read-only SNMP group is configured with the name ADMIN, it is set to SNMPv3 with authentication and encryption required, and only allows access to the host permitted in the SNMP-NMS ACL.

Step 4

Configure a SNMPv3 user, assign it to the ADMIN group, and use SHA-1 and AES-128 for authentication and encryption.

```
rtr01(config)# snmp-server user USER1 ADMIN v3 auth sha cisco12345 priv aes 128
cisco54321
rtr01(config)#
Jan  1 01:52:49.896: Configuring snmpv3 USM user, persisting snmpEngineBoots. Please Wait...
```

Once the user is created, you should get a message at the console indicating that the internal SNMP process is starting up.

Step 5

Configure a SNMPv3 trap receiver at 192.168.100.100 and assign the user created in Step 4.

```
rtr01(config)# snmp-server host 192.168.100.100 version 3 priv USER1
```

Step 6

Enable SNMP trap sending to send a trap to the NMS server when an interface on the router goes down or up.

```
rtr01(config)# snmp-server enable traps snmp linkdown linkup
```

Step 7

Configure interface index persistence.

```
rtr01(config)# snmp-server ifindex persist
```

Step 8

Save your configuration to NVRAM and reload the rtr01 router.

```
rtr01# copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
rtr01# reload
Proceed with reload? [confirm]
```

> Note: Because of a bug in the Cisco IOL image in CML, you need to reboot the router after starting the SNMPv3 process to ensure that the next steps work correctly. Configuration of SNMPv3 in a production network does NOT require a device reboot.

Step 8

After the router reboots and you can log back into Privileged EXEC mode, start a Packet Capture on the link between rtr01 and the NMS server and filter the capture to UDP port 162.

Step 9

Verify SNMP configuration

To verify that traps are being sent, issue the command `debug snmp packets` and shutdown the E0/0 interface and then re-enable it. You should see debug output indicating that two SNMP packets were sent. It might take a few moments for the router to start sending traps and for them to appear in the packet capture.

```
rtr01# debug snmp packets
SNMP packet debugging is on
rtr01# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
rtr01(config)# interface e0/0
rtr01(config-if)# shutdown
rtr01(config-if)#
Apr 26 01:10:18.534: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to administratively down
Apr 26 01:10:19.534: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to down
rtr01(config-if)# no shutdown
rtr01(config-if)#
Apr 26 01:10:20.534: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
rtr01(config-if)#
Apr 26 01:10:21.534: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to up
rtr01(config-if)#
Apr 26 01:10:22.534: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address 192.168.255.184, mask 255.255.255.0, hostname rtr01
rtr01(config-if)#
Apr 26 01:10:23.534: SNMP: Queuing packet to 192.168.100.100
Apr 26 01:10:23.534: SNMP: V2 Trap, reqid 1, errstat 0, erridx 0
 sysUpTime.0 = 754478
 snmpTrapOID.0 = snmpTraps.3
 ifIndex.1 = 1
 ifDescr.1 = Ethernet0/0
 ifType.1 = 6
 lifEntry.20.1 = administratively down
Apr 26 01:10:24.534: SNMP: Queuing packet to 192.168.100.100
Apr 26 01:10:24.534: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
 sysUpTime.0 = 754479
 snmpTrapOID.0 = snmpTraps.4
 ifIndex.1 = 1
 ifDescr.1 = Ethernet0/0
 ifType.1 = 6
 lifEntry.20.1 = up
Apr 26 01:10:25.534: SNMP: Packet sent via UDP to 192.168.100.100
Apr 26 01:10:25.534: SNMP: Packet sent via UDP to 192.168.100.100
```

Notice that in the output above, two traps were sent after the interface was brought back up. Your results might differ but you should still get two traps captures in the Packet Capture window.

Disable debugging with the `do undebug all` command.

Step 10

Decrypt the SNMPv3 PCAP using Wireshark.

Notice that the `Info` column of the Packet Capture window shows that the PDU is encrypted and that the private key is unknown. You will need to download the PCAP from CML and use Wireshark to decrypt the SNMPv3 packet.

Click the `Download` button in the Packet Capture window and open the PCAP in Wireshark on your PC.

Right-click one of the SNMPv3 packets displayed in the top window in Wireshark, select `Protocol Preferences` from the menu, select `Simple Network Management Protocol` and then select `Users Table`.

Click the `+` button to create a new user entry. Enter the following values:

User: USER1 Authentication Model: SHA-1 Password: cisco12345 Privacy Protocol: AES Privacy password: cisco54321

Click `Ok` to save the USER1 table entry.

Fully expand the `msgData: encryptedPDU` entry. In the last expanded field, you will see information relating to the Ethernet 0/0 interface ("administratively down" or "up" depending on the SNMPv3 packet you have selected.

Step 11

Verify the SNMPv3 configuration on rtr01.

```
rtr01# show snmp
Chassis: 131184943
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Set-request PDUs
    0 Input queue packet drops (Maximum queue size 1000)
    0 Dispatcher queue packet drops (Maximum queue size 75)
2 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
    2 Trap PDUs
Packets currently in SNMP process input queue: 0, max 1000
Packets currently in SNMP PDU dispatcher queue: 0, max 75
SNMP global trap: disabled

SNMP logging: enabled
    Logging to 192.168.100.100.162, 0/10, 2 sent, 0 dropped.

rtr01# show snmp group
groupname: ADMIN                            security model:v3 priv
contextname:          storage-type: nonvolatile
readview : SNMP-RO                          writeview:
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F
row status: active      access-list: SNMP-NMS

rtr01# show snmp user

User name: USER1
Engine ID: 800000090300AABBCC012F00
storage-type: nonvolatile        active
Authentication Protocol: SHA
Privacy Protocol: AES128
Group-name: ADMIN
```

Step 12

Install SNMP and SNMP tools on the SNMP-NMS device and perform a SNMP walk on the rtr01 device.

`snmpwalk` is a command-line utility used to retrieve a subtree of management values from a network device using SNMP GETNEXT requests. It's essentially a tool that automates the process of querying multiple SNMP-enabled devices for information, instead of having to query each device individually. This allows you to gather a wide range of data from devices like routers, switches, and other network infrastructure.

```
snmp_nms:~$ sudo apk add net-snmp
fetch http://dl-cdn.alpinelinux.org/alpine/v3.14/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.14/community/x86_64/APKINDEX.tar.gz
(1/4) Installing net-snmp-libs (5.9.3-r1)
(2/4) Installing net-snmp-agent-libs (5.9.3-r1)
(3/4) Installing net-snmp (5.9.3-r1)
(4/4) Installing net-snmp-openrc (5.9.3-r1)
Executing busybox-1.33.1-r8.trigger
OK: 1215 MiB in 334 packages

snmp_nms:~$ sudo apk add net-snmp-tools
(1/1) Installing net-snmp-tools (5.9.3-r1)
Executing busybox-1.33.1-r8.trigger
OK: 1216 MiB in 335 packages

snmp_nms:~$ snmpwalk -v3 -u USER1 -l authPriv -a SHA -A "cisco12345" -x AES -X "cisco54321" 192.168.100.1 1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Cisco IOS Software [Dublin], Linux Software (X86_64BI_LINUX-ADVENTERPRISEK9-M), Version 17.12.1, RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2023 by Cisco Systems, Inc.
Compiled Thu 27-Jul-23 22:33 by m
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.1
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1072907) 2:58:49.07
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: rtr01.example.com
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 78
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

In the `snmpwalk` command above, you must specify the following parameters:

- SNMP version
- the user name
- the level of securty (auth and priv)
- the level of SHA authentication and the password
- the level of AES encryption and the password
- the IP address of the device to check (rtr01)
- the OID your want to view (in this case, the router's system information)

In the example above, the OID specified pulls system and IOS version information from a device.

Use the `snmpwalk` command and explore other OID values, like 1.3.6.1.2.1.4.24 (routing table) or 1.3.6.1.2.1.2.2.1.8 (operational status of the devices interfaces.)

Congratulations! You have completed the lab. You learned how to configure and verify SNMPv2c and SNMPv3, as well as use Wireshark and `snmpwalk` to analyze and gather OIDs from Cisco routers and switches.