sCSCI 5180 – Network Management

and Automation

Lab 1

Network management using SNMP and NMAP

University of Colorado Boulder

Department of Computer Science

Professor Levi Perigo, Ph.D.

# Summary

SNMP is used widely by network and system administrators to monitor the health and metrics of a diverse array of network devices.

The objectives in this lab will enable you to understand how different SNMP versions work, gather operational statistics and monitor your network using simple commands, and modify parameters remotely on SNMP agents.

# Pre-Lab

You will need the following commands to enable SNMP on the Cisco router in the VM's GNS3. (Note: Use the instructions from Lab 0 for gaining access to the VM and GNS3 setup.)

- Run the simulation by clicking on the Play button in GNS3.  
  
- Console into the router, check if SNMP is running using **show snmp host**.

If SNMP is not enabled, follow these steps to configure SNMP host on a Cisco router:  
  
- Enable SNMP traps on the router by entering: (config)#**snmp-server enable traps**   
  
- Assign an IP address (make sure it is in a different subnet than the primary interface, use any private subnet) to the 2nd interface of the router that you added & bring the interface up.

- Enter configuration commands, one per line. End with CNTL/Z.

(config)# **snmp-server host 198.51.100.2 public**

(config)# **snmp-server community public rw**

\*Note: The "snmp-server host" IP address is the IP address of the VM terminal. Thus, in this example the IP address would be 198.51.100.2.

On the terminal of the VM start Wireshark and monitor the tap0 interface.

Next type the below commands in the VM terminal and check the output (you can receive SNMP data from the router using **SNMPGET/SNMPWALK**).

netman@netman:~$ snmpget -v 1 -c public 198.51.100.3 ifName.1

IF-MIB::ifName.1 = STRING: Fa0/0 ------( This is the output )

netman@netman:~$ snmpget -v 1 -c public 198.51.100.3 .1.3.6.1.2.1.2.1.0

IF-MIB::ifNumber.0 = INTEGER: 5 ------( This is the output )

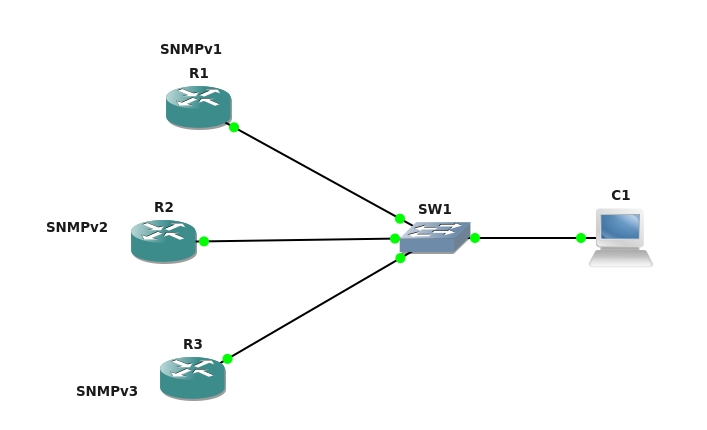
\*NOTE: The IP address used within the terminal is the IP address of the Cisco router. In this example the Cisco router has the IP address of 198.51.100.3.

You should be able to see a similar output on the terminal as well as an SNMP packet on Wireshark.

# Objective 1: Configuring SNMP on Cisco IOS

Create the topology in GNS3 as shown below and assign management IPs (198.51.100.0/24 subnet) to them on fa0/0. Configure the nodes for different versions of SNMP & enable traps.

- R1: SNMPv1 (Already configured)  
- R2: SNMPv2   
- R3: SNMPv3



1. How did you configure SNMPv2 and v3 on routers R2 and R3? Provide running configuration screenshots (only portions relevant to SNMP). [**10 points**]
   1. How I configured SNMPv2 on router2 is that I set the interface IP to be in the same subnet as the VM. I then enabled traps and configured a community string for RO and specifying the host as the IP of the virtual machine with version of 2c. Screenshots below for context:

A close up of white text

Description automatically generated

* 1. For SNMPv3, it was a bit harder. I created a view, group, and user. I also enabled traps and set an IP for the router in the same subnet as the snmp user. Screenshots below for context:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated



# Objective 2: SNMPGET and Dashboard

The list of OIDs that need to be fetched from the routers:

sysContact = 1.3.6.1.2.1.1.4.0

sysName = 1.3.6.1.2.1.1.5.0

sysLocation = 1.3.6.1.2.1.1.6.0

ifNumber = 1.3.6.1.2.1.2.1.0

sysUptime = 1.3.6.1.2.1.1.3.0

Sample command to run on terminal:

**snmpget -v 1 -c public 198.51.100.3 .1.3.6.1.2.1.1.4.0**

1. Enter the above SNMPGET commands for the OIDs mentioned for SNMP v1, v2, and v3. Paste relevant screenshots. [**10 points**]

* For SNMPv1:

A screenshot of a computer program

Description automatically generated

* For SNMPv2:

A screen shot of a computer program

Description automatically generated

* For SNMPv3:

A black screen with white text

Description automatically generated

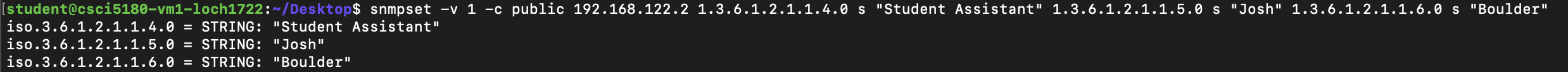
1. Create a dashboard to display the output from those commands using UNIX/Python. Paste relevant screenshots. [**15 points**]

A screenshot of a computer program

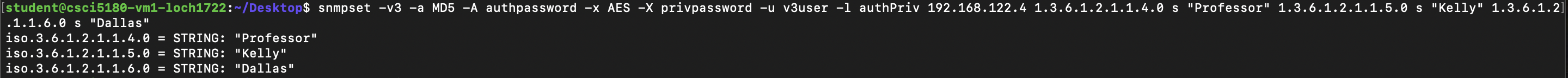
Description automatically generatedA screenshot of a computer

Description automatically generated

1. Use SNMPSET commands to modify Contact, Name, and Location to display varied output for each version: 1 and 2. Paste relevant screenshots. [**10 points**]

****

****

****

**A screen shot of a computer

Description automatically generated**

**Sample dashboard to be displayed using UNIX/Python:**

**SNMP v1**

Contact: Student Assistant

Name: Josh

Location: Boulder

Number: 2

Uptime: 0:54:20.47

**SNMP v2**

Contact: Student

Name: George

Location: San Diego

Number: 2

Uptime: 0:67:10.57

**SNMP v3 (any of the 2)**

Contact: Professor

Name: Kelly

Location: Dallas

Number: 2

Uptime: 1:24:20.47

# Objective 3: SNMPSET Commands

**NOTE:** Must use SNMPSET commands to perform the below tasks on Router 1 in GNS3:

1. Change the hostname to “**csci-7000-10**” (provide a screenshot) [**10 points**]  
   
2. Change the interface status of the secondary interface (NOT THE MANAGEMENT INTERFACE) to “**Up**” (Assuming it’s up, if not, change to “**Admin Down**”). Provide screenshots. [**10 points**]  
   A screenshot of a computer screen

   Description automatically generated
3. Create a SNMP contact profile with the name (provide a screenshot): <**yourname@colorado.edu**> [**10 points**]





Objective 4: SNMP Traps and Wireshark/TCPDUMP

1. Start a new Wireshark capture on the tap0 interface of the VM. Apply a display filter to filter SNMP traffic.
2. Shutdown the interfaces on R2 and R3, and bring them up again. Do you observe different trap messages being exchanged between the SNMP agent and the manager (VM) in the packet capture? Provide relevant screenshots. [**10 points**]

* I do observe different trap messaging being exchanged. It seems that the SNMPv2 traps are in plain text and the SNMPv3 traps are encrypted. The first screenshot shows a snippet of SNMPv2 trap and the second secreenshot shows the SNMPv3 encrypted trap.



1. Start a capture using TCPDUMP. Bring down an interface on any of the routers (this should generate a trap). Store the output in a .pcap file. After stopping the TCPDUMP, create a Python script that will analyze and parse the .pcap file for a Trap. Then the Python script should generate an email, to your email id, with the contents of the Trap [<https://www.pythonforbeginners.com/google/sending-emails-using-google>]. Provide relevant screenshots and submit the code. [**20 points**]

A computer screen shot of a program

Description automatically generatedA screenshot of a computer

Description automatically generated

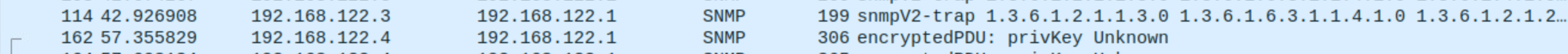
1. What are the key differences you can observe between the trap messages for SNMPv2 and v3? Provide relevant screenshots highlighting the differences. [**10 Points**]

The main key difference I notice is the v2 is in plain text and v3 is encrypted. Also, in the v2 trap, certain values are shown like the OIDs of the trap. But in v3, certain msg\_\_\_\_ messages are shown that contain data but are encrypted. Screenshots included below for understanding:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated  


Objective 5: Network Administration using SNMP [Extra Credit]

Imagine a Data Center or Service Provider network. You, being a principle network engineer, get a ticket for eBGP sessions going down on multiple routers. You start analyzing the output of all the possible “show” commands in BGP that you are aware of. However, all configurations and parameters look perfect and you scratch your head for a while trying to know the root cause of the issue. You run down to the data center/lab and check all the physical connections. On doing a “show ip interface brief” on all the affected routers, you see that some of the interfaces have been taken down administratively and the others show a Protocol down. Most networking problems reside at the lower levels and hence troubleshooting layer 1 is the first step of a bottom-up approach. The following objective will help you find an easier and faster way to check the layer 1 status before moving up the OSI model for troubleshooting. (**12 points**)

1. Configure descriptions for the router interfaces for easier administration (e.g. Router(config-if)# description Management Interface).
2. Write a script in a language of your choice (e.g. UNIX/Python) to extract and display interface information from all the routers in the above topology using the following MIB objects (Hint: you can view entire MIB details using SNMPBULKWALK command).

* ifName
* ifDescr
* ifOperStatus
* iPhysAddress
* ifAdminStatus
* ifInUcastPkts

**Sample output to be displayed by the script:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Interface Name | Description | Operational Status | Physical Address | Admin Status | Incoming Unicast Packet Counter |
| R1 | Fa0/0 | Management Interface | Up | 00-03-47-92-9C-6F | Up | 100 |

Provide relevant screenshots.

1. Modify the above script to retrieve and display: interface IP address and network mask information. Provide relevant screenshots.
2. Implement both the scripts (TCPDUMP Trap obj 4.3 and extract interface info obj 5.2) using just one script. Also, ensure your script shall continuously monitor the interface status, display the interface information (as in obj 5.2) and parse the trap (as in obj 4.3). Provide relevant screenshots.

Report Questions (5 points each)

1. Would you recommend using a management subnet for SNMP? Why/why not?
   * I would recommend using a management subnet for SNMP because it allows for isolating the devices that are meant to be managed in its own separate network. Also, reduces the amount of traffic on a network that would be shared.
2. Why is a switch used in the network design in GNS3?
   * A switch is used in the network design in GNS3 so that all the routers are in the same domain. This allows the snmp host to easily manage these devices because they are in the same network.
3. Can you use a router instead? Why/why not?
   * You could possibly use a router instead if it had enough ports. You would also have to allow for traps sent and messages sent from the host to retain the same source IP so that it is shown as coming from that specific device.
4. If you used a router, what would need to change (if anything).
   * As mentioned before, adding interfaces and forwarding messages with the same source IP.
5. What command has to be entered on the router, to disable configuration changes to be made through SNMP?
   * Snmp-server chassis-id none

Network Discovery using NMAP

Objectives

* Learn the basic operations of network discovery using Nmap.
* Learn how to capture and analyze ICMP traffic.
* Learn how to capture and analyze port scanning traffic.
* Perform IP address spoofing.
* Gather OS information.
* Perform Scripting and Automation.

Summary

Nmap is a free open source tool that can be used for performing a variety of network scanning and security functions. To create a “map” of the network, Nmap sends specific packets to the target host (or hosts) and then analyzes the responses. Nmap can also be used to enumerate networks and avoid IDS through spoofing/stealth, please use this responsibly and follow the lab directions.

Nmap is available for download for many Linux distributions (There is also a version available for Windows). It also comes with a GUI (Zmap) that can be used as an alternative to the CLI. The functions of this lab will focus on ping sweeps (find hosts), port scanning (determine vulnerabilities/services), IP spoofing (avoiding detection by IDS), and gathering intelligence on a network.

Objective 1: Download and Install Nmap/Zmap on Your Machine

Follow the instructions from the Nmap website for your operating system:

<https://nmap.org/>

For the remainder of this lab, you can use **Nmap or Zenmap**

Objective 2**:** Ping Sweeps and Port Scans

1. Perform a ping sweep for the following network (Note: this only works from CU

network or VPN; if unavailable use your home/private network): **172.20.74.0/24**

1. Provide a screenshot showing the command and the results [**5 points**]

A computer screen with white text

Description automatically generated

1. How many devices responded to the ping sweep? Provide information about how you can determine this. [**2.5 points**]

5 devices responded to the ping sweep. I can determine this by looking at the bottom of the nmap where it says how many devices were scanned.

2. Choose a host that replied from the ping sweep; now perform a full scan on that host

a. Which well-knownports were open on this machine? Provide the screenshot. [**2.5 points**]

port 53 was open on this machine (DNS). A screen shot of a computer

Description automatically generated

1. Provide the command you would use to perform a “stealth”scan. [**2.5 points**]

Nmap -sS <target IP>

Objective 3: IP Spoofing and OS Detection

1. Perform a full network scan on the /24 network (optional: use a spoofed IP address (use target IP address from previous objective as the source))

a. Provide the command used [**2.5 points**]

nmap 172.20.74.0/24

b. Explain the different “state”options for a Nmap port scan (i.e. open, filtered, closed, etc.) [**2.5 points**]

open: port is open and accepting connections.

closed: not accepting connections and port is closed

filtered: can’t determine if port is open/closed

unfiltered: port state is determined

open|filtered: unable to determine if open or filtered

closed|filtered: unable to determine if closed or filtered

2. Provide screenshots of the Operating Systems running on each of these

machines [**2.5 points**]

A screenshot of a computer

Description automatically generated

Objective 4: Scripting and Automation

1. IP Address Mapping

a. If using the VM, install Nmap

**#sudo apt-get install nmap**

b. Run a ping sweep on the /24 network

c. Using **Bash or Python**, record the IP addresses into a **text/CSV** file d. Repeat the ping sweep after some time (~10 min.)

e. Compare the two files

i. Were there any differences? If so, what is different? [**2 points**]

The only differences I saw in the NMAPs was the latency at which the host responded to and the amount of time it took to do the nmap.

A computer screen shot of a black screen

Description automatically generated

ii. Submit the scripts, files, procedures or screenshots of how you accomplished this [**10 points**]

nmap -oN nmap.txt -sn 172.20.74.0/24 && sleep 600 && nmap -oN nmap2.txt -sn 172.20.74.0/24 && diff nmap.txt nmap2.txt

f. As a network manager, list one thing that is useful and one thing could be detrimental with this information [**5 points**]

One thing that could be useful is knowing what devices are on the network and what ports are open. From there you could determine what devices should be doing their jobs. Something detrimental with this information is someone using a spoofed IP address can scan the network and see all of the devices connected to a network. They can understand what devices are doing what and on what IP.

2. **Extra Credit:**

Rogue Web Server (web servers ending with IP addresses .1-.10 are legitimate;

outside of that range are rogue)

a. Run a full network port scan to find open ports for **80, 443**, and **8080**

b. Submit the file of all web servers that are not in the range (i.e. rogue web server)

i. How did you accomplish this? [**5 points**]

Report Questions

1. How can you set a decoy, to hide your source IP address using Nmap? [**2.5 points**]

Nmap -D decoy target

2. List some ways Nmap can be used to trick a firewall. [**2.5 points**]

Using the -O can display the OS of the devices and can potentially determine the type of firewall in use

Also using the -D decoy flag can obscure where the nmap is coming from

The -sV can scan for versions of services running on the open ports. This can give the intruder information

# Total Score = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ 167 [+17 Bonus]