Network Management and Automation

Lab 1
Network management using SNMP and NMAP

University of Colorado Boulder
Network Engineering Program

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)

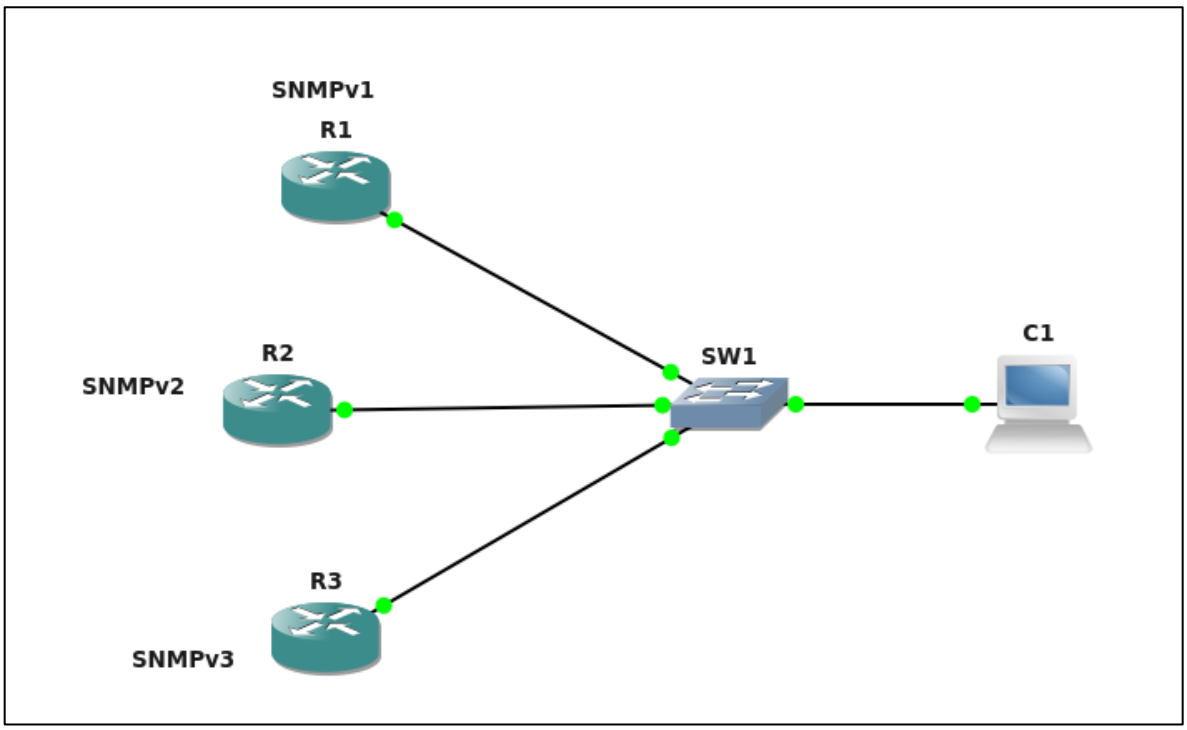
*NOTE: The IP address used within the terminal is the IP address of the Cisco router. In this example the Cisco router has the IP address of 198.51.100.3.

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**]
   Answer:
   Below is the screenshot of SNMPv2 configuration:

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#snmp-server community SUNNY ro
R2(config)#snmp-server location BOULDER
R2(config)#snmp-server contact sunnybajaj627@gmail.com
R2(config)#snmp-server host 198.51.100.2 version 2c SUNNY
R2(config)#snmp-
R2(config)#snmp-server enable traps
R2(config)#exit
R2#show snmp host
Notification host: 198.51.100.2 udp-port: 162   type: trap
user: SUNNY     security model: v2c
```

Below is the screenshot of SNMPv3 configuration:

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int f0/0
R3(config-if)#ip address 198.51.100.6 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#
*Jan 23 02:22:10.283: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan 23 02:22:11.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to
up
R3(config-if)#exit
R3(config)#int f1/0
R3(config-if)#ip address 198.51.101.2 255.255.255.0
R3(config-if)#no shut
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*Jan 23 02:22:52.907: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
*Jan 23 02:22:53.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to
up
R3(config)#snmp-
R3(config)#snmp-server enable traps
R3(config)#snmp-server host 198.51.100.2 v3
R3(config)#snmp-server community public rw
R3(config)#
```

```
R3(config)#snmp-server enable traps
R3(config)#snmp-
R3(config)#snmp-server host 198.51.100.2 v3
R3(config)#snmp-server community public rw
R3(config)#
```

```
R3(config)#snmp-server group GALAXY v3 priv
R3(config)#snmp-
R3(config)#snmp-server user ?
  WORD  Name of the user

R3(config)#snmp-server user MYGALAXY GALAXY v3 ?
  access     specify an access-list associated with this group
  auth       authentication parameters for the user
  encrypted  specifying passwords as MD5 or SHA digests
  <cr>

R3(config)#snmp-server user MYGALAXY GALAXY v3 auth ?
  md5  Use HMAC MD5 algorithm for authentication
  sha  Use HMAC SHA algorithm for authentication

R3(config)#snmp-server user MYGALAXY GALAXY v3 auth md5 SUNNY123 ?
  access  specify an access-list associated with this group
  priv    encryption parameters for the user
  <cr>

R3Wireshark#$ user MYGALAXY GALAXY v3 auth md5 SUNNY123 priv aes 128 BAJAJ123
R3(config)#
*Jan 22 09:51:38.258: Configuring snmpv3 USM user, persisting snmpEngineBoots. Please Wait...
```

```
R3#show running-config | incl snmp
snmp-server group GALAXY v3 priv
```

```
R3#show snmp user

User name: MYGALAXY
Engine ID: 800000090300CA0347C20000
storage-type: nonvolatile        active
Authentication Protocol: MD5
Privacy Protocol: AES128
Group-name: GALAXY
```

```
R3#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: GALAXY                              security model:v3 priv
contextname: <no context specified>            storage-type: nonvolatile
readview : v1default                           writeview: <no writeview specified>
notifyview: <no notifyview specified>
row status: active

groupname: public                              security model:v1
contextname: <no context specified>            storage-type: permanent
readview : v1default                           writeview: v1default
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F
row status: active

groupname: public                              security model:v2c
contextname: <no context specified>            storage-type: permanent
readview : v1default                           writeview: v1default
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F
row status: active
```

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**]

   Answer:

   Below is the screenshot of all the SNMPGET commands for the OID mentioned above for SNMPv1:

```
netman@netman:~$ snmpget -v1 -c public 198.51.100.3  .1.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING: suba3747@colorado.edu
netman@netman:~$ snmpget -v1 -c public 198.51.100.3  .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: R1
netman@netman:~$ snmpget -v1 -c public 198.51.100.3  .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING: BOULDER
netman@netman:~$ snmpget -v1 -c public 198.51.100.3  .1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 7
netman@netman:~$ snmpget -v1 -c public 198.51.100.3  .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (6091720) 16:55:17.20
netman@netman:~$
```

   Below is the screenshot of all the SNMPGET commands for the OID mentioned above for SNMPv2:

```
netman@netman:~$ snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING: suba3747@colorado.edu
netman@netman:~$ snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: R2
netman@netman:~$ snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING: BOULDER
netman@netman:~$ snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 7
netman@netman:~$ snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (6061320) 16:50:13.20
netman@netman:~$
```

Below is the screenshot of all the SNMPGET commands for the OID mentioned above for SNMPv3:

```
netman@netman:~$ snmpget -v3 -u MYUSER -l AuthPriv -a md5 -A MYPASS123 -x aes -XMYKEY123 198.51.100.6 .1
.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING: suba3747@colorado.edu
netman@netman:~$ snmpget -v3 -u MYUSER -l AuthPriv -a md5 -A MYPASS123 -x aes -XMYKEY123 198.51.100.6 .1
.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: R3
netman@netman:~$ snmpget -v3 -u MYUSER -l AuthPriv -a md5 -A MYPASS123 -x aes -XMYKEY123 198.51.100.6 .1
.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING: BOULDER
netman@netman:~$ snmpget -v3 -u MYUSER -l AuthPriv -a md5 -A MYPASS123 -x aes -XMYKEY123 198.51.100.6 .1
.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 7
netman@netman:~$ snmpget -v3 -u MYUSER -l AuthPriv -a md5 -A MYPASS123 -x aes -XMYKEY123 198.51.100.6 .1
.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (2720477) 7:33:24.77
netman@netman:~$
```

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

Answer:

Displaying the commands using Python:

SNMPv2:

```
snmpv1.py - Visual Studio Code
dit  Selection  View  Go  Run  Terminal  Help
smtp.py   gui.py   snmpv1.py ×   snmpv2.py   sunny.py
home > netman > Desktop > snmpv1.py
#!/usr/bin/env python3

import os

os.system("snmpget -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.4.0")
os.system("snmpget -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.5.0")
os.system("snmpget -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.6.0")
os.system("snmpget -v1 -c public 198.51.100.3 .1.3.6.1.2.1.2.1.0")
os.system("snmpget -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.3.0")

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
netman@netman:~$ /usr/bin/python3 /home/netman/Desktop/snmpv1.py
SNMPv2-MIB::sysContact.0 = STRING: suba3747@colorado.edu
SNMPv2-MIB::sysName.0 = STRING: R1
SNMPv2-MIB::sysLocation.0 = STRING: BOULDER
IF-MIB::ifNumber.0 = INTEGER: 7
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (235129) 0:39:11.29
netman@netman:~$
```

SNMPv3:

snmpv2.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

smtp.py | gui.py | snmpv1.py | snmpv2.py | sunny.py

home > netman > Desktop > snmpv2.py

```python
#!/usr/bin/env python3

import os

os.system("snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.4.0")
os.system("snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.5.0")
os.system("snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.6.0")
os.system("snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.2.1.0")
os.system("snmpget -v2c -c public 198.51.100.4 .1.3.6.1.2.1.1.3.0")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
netman@netman:~$ /usr/bin/python3 /home/netman/Desktop/snmpv2.py
SNMPv2-MIB::sysContact.0 = STRING: suba3747@colorado.edu
SNMPv2-MIB::sysName.0 = STRING: R2
SNMPv2-MIB::sysLocation.0 = STRING: BOULDER
IF-MIB::ifNumber.0 = INTEGER: 7
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (201311) 0:33:33.11
netman@netman:~$
```

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

Answer:

Below is the screenshot of SNMPSET commands to modify the Contact, Name & Location:

SNMPv1

```
netman@netman:~$ snmpset -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.4.0 s sunnybajaj627@gmail.com
SNMPv2-MIB::sysContact.0 = STRING: sunnybajaj627@gmail.com
netman@netman:~$ snmpset -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.5.0 s SUNNY
SNMPv2-MIB::sysName.0 = STRING: SUNNY
netman@netman:~$ snmpset -v1 -c public 198.51.100.3 .1.3.6.1.2.1.1.6.0 s INDIA
SNMPv2-MIB::sysLocation.0 = STRING: INDIA
netman@netman:~$
```

SNMPv2:

```
netman@netman:~$ snmpset -v2c -c public 198.51.100.3 .1.3.6.1.2.1.1.4.0 s sunnybajaj627@gmail.com
SNMPv2-MIB::sysContact.0 = STRING: sunnybajaj627@gmail.com
netman@netman:~$ snmpset -v2c -c public 198.51.100.3 .1.3.6.1.2.1.1.5.0 s SUNNY
SNMPv2-MIB::sysName.0 = STRING: SUNNY
netman@netman:~$ snmpset -v2c -c public 198.51.100.3 .1.3.6.1.2.1.1.6.0 s INDIA
SNMPv2-MIB::sysLocation.0 = STRING: INDIA
netman@netman:~$
```

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**]
   Answer:
   Below is the screenshot of the hostname:

```
netman@netman:~$ snmpset -v1 -c public 198.51.100.3  .1.3.6.1.2.1.1.5.0 s csci-7000-10
SNMPv2-MIB::sysName.0 = STRING: csci-7000-10
netman@netman:~$
```

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**]
   Answer:
   Below is the screenshot of the interface:

```
netman@netman:~$ snmpset -v1 -c public 198.51.100.3 IF-MIB::ifAdminStatus.2 i 2
IF-MIB::ifAdminStatus.2 = INTEGER: down(2)
netman@netman:~$
```

3. Create a SNMP contact profile with the name (provide a screenshot): <**yourname@colorado.edu**> [**10 points**]
   Answer:
   Below is the screenshot of the Contact Profile:

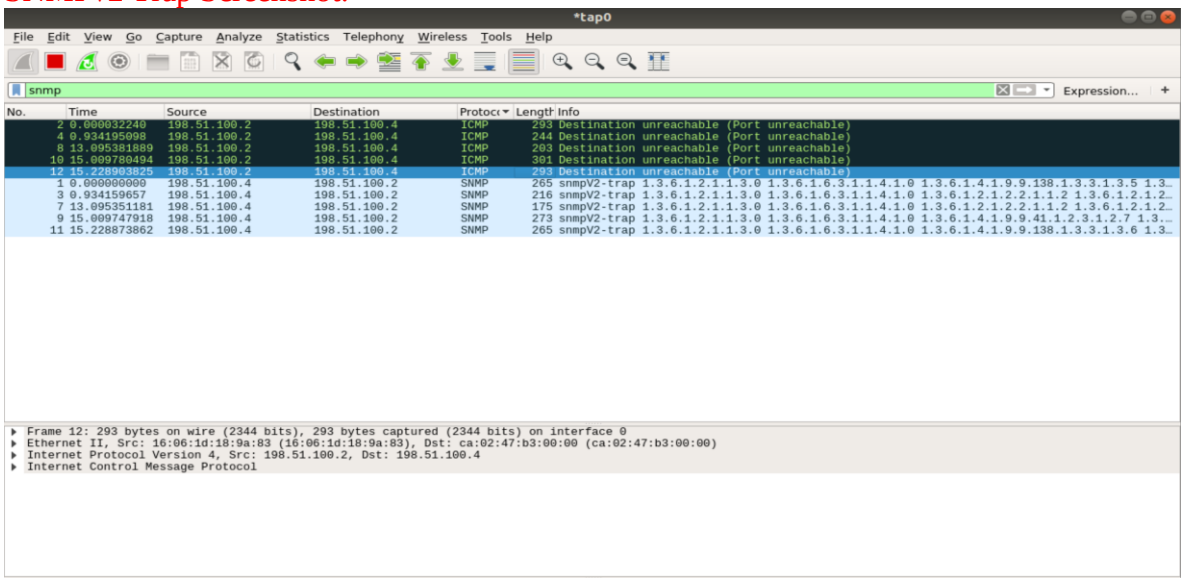
```
netman@netman:~$ snmpset -v1 -c public 198.51.100.3  .1.3.6.1.2.1.1.4.0 s suba3747@colorado.edu
SNMPv2-MIB::sysContact.0 = STRING: suba3747@colorado.edu
netman@netman:~$
```

Objective 4: SNMP Traps and Wireshark/TCPDUMP

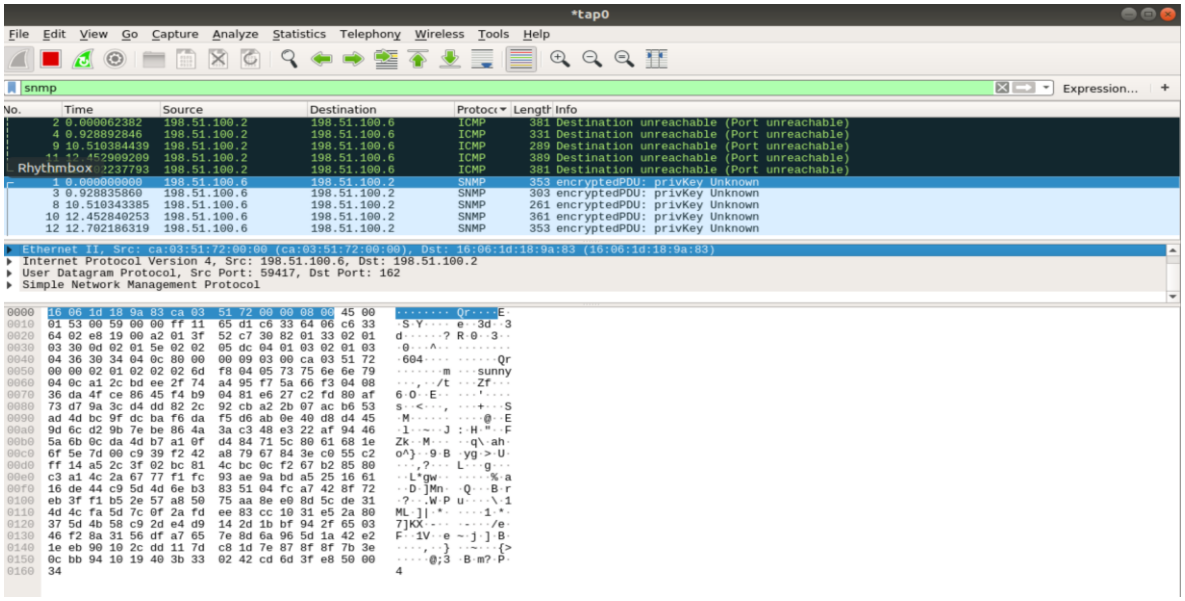
1. Start a new Wireshark capture on the tap0 interface of the VM. Apply a display filter to filter SNMP traffic.

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**]
   Answer:
   Below is the screenshot of the different trap messages being exchanged between the SNMP agent and the manager:
   SNMPv2 and SNMPv3 Traps Information:
   The IP address 198.51.100.4 is assigned to Router 2 (SNMP agent) and the IP address 198.51.100.6 (SNMP agent) is assigned to Router 3 and the traps for SNMPv3 and SNMPv2 you can see in the below diagram.

   SNMPv2 Trap Screenshot:



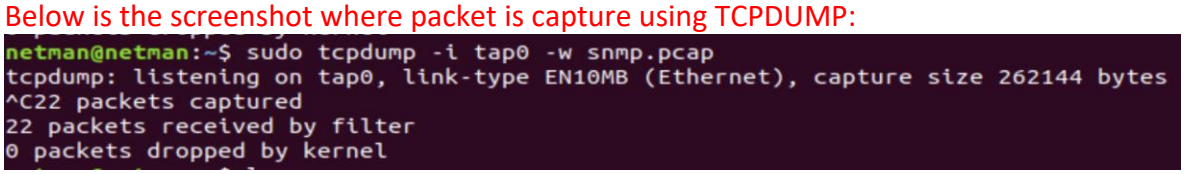
SNMPv3 Trap Screenshot:



3. 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**]

   Answer:

   Below is the screenshot where packet is capture using TCPDUMP:

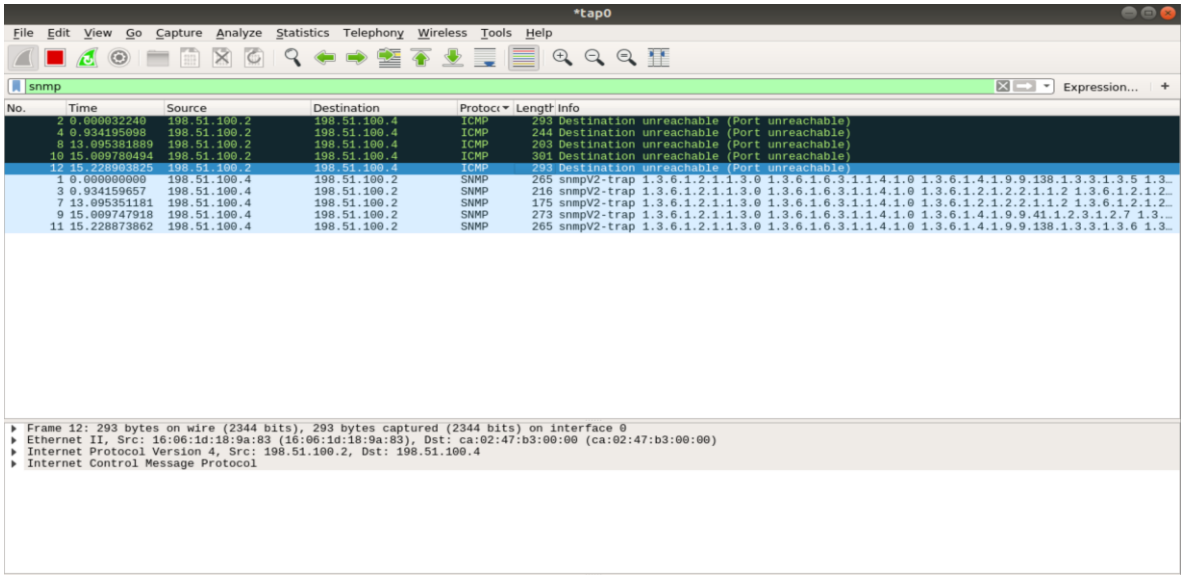


   Also, I have attached the pcap file.

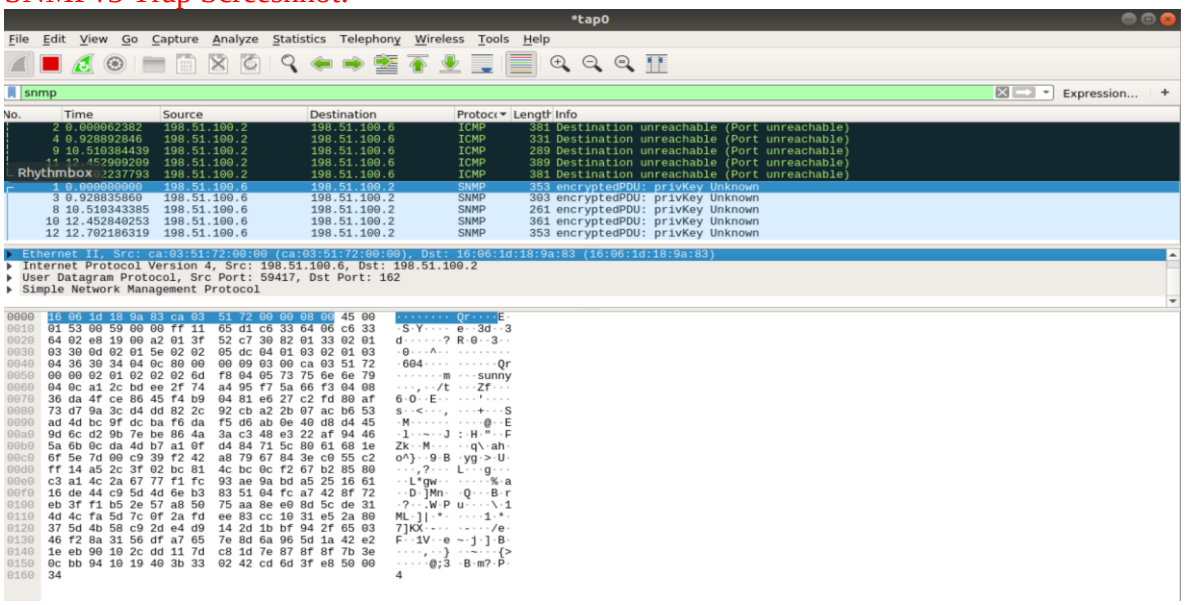
4. What are the key differences you can observe between the trap messages for SNMPv2 and v3? Provide relevant screenshots highlighting the differences. [**10 Points**]

   Answer:

   SNMPv2 Trap Screenshot:



SNMPv3 Trap Screesnhot:



The main difference between the SNMPv2 and SNMPv3 is that SNMP version 3 provides security and SNMP version 2 doesn't provide security. As you can see in the above diagram where SNMP version 2 trapped is shown there is no security and in SNMP version 3 it shows encryptedPDU which is security.

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.

3. Modify the above script to retrieve and display interface IP address and network mask information. Provide relevant screenshots.

4. 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?

   Answer:

   Yes, we can recommend using a management subnet for SNMP. If we use management subnet, we can able to get trapped message by opening the VM terminal

and shut the other interface. By doing so you can be able to see the trapped message in Wireshark. As in the above objective 4, we have executed.

2. Why is a switch used in the network design in GNS3?
   Answer:
   Switches are the essential elements when it comes to designing any network. They are used to connect multiple machines like printers, servers, personal computers, laptops, wireless access points on the local area network within an organization or campus. With the help of a switch, you can share the data with the connected devices that the switch has enabled.
3. Can you use a router instead? Why/why not?
   Answer:
   We cannot use the router instead of the switch. Router is used when we convey the information to the different network.
   For instance, Let's say that we have a office in Denver for our company. In that office I will be having devices like PCS, Server and Printer and they need to be able to communicate with each other. So, for that will connect network switch with the devices by Ethernet cable. A switch is what allows the connectivity on our local area network. In that office I also have Laptop which is going to connect over a wireless network. So, for that we need wireless access point which is going to be connected to switch. sI want my end host to communicate with other devices as well on the Internet (Wide Area Network). Now, Denver office wants to communicate with the New York office. So, for that I will use an advanced device which is router. A router can make advanced routing decision to route traffic between different areas of the network.
4. If you used a router, what would need to change (if anything).
   Answer: Yes, I have used a router in which I have done the SNMPv1, SNMPv2 and SNMPv3 configuration which is used between the SNMP agent and NMS. Also, I have configured a private subnet on different subnet to get the trap message from SNMPv2 and SNMPv3.
5. What command has to be entered on the router, to disable configuration changes to be made through SNMP?
   Answer:
   To disable the configuration which you have made on router, you must type "no".
   For Instance, you have done any configuration by the following command: snmp-server community sunny ro. So, to disable it we can use the following command: no snmp-server community sunny ro.

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

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

      Answer:

      Below is the screenshot:

```
netman@netman:~$ nmap 10.0.0.131-255

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-23 23:04 MST
Nmap scan report for 10.0.0.255
Host is up (0.0024s latency).
Not shown: 999 closed ports
PORT     STATE     SERVICE
514/tcp filtered shell

Nmap done: 125 IP addresses (1 host up) scanned in 25.93 seconds
netman@netman:~$
```

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

Answer:

By running above command in question 1a only one device responded to the ping sweep.

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

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

Answer:

Only one ports were open on the machine which is 514/TCP.

c. Provide the command you would use to perform a "stealth" scan. [**2.5 points**]

Answer:

```
netman@netman:~$ sudo nmap -sS -P0 172.20.74.220

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-23 23:46 MST
Nmap scan report for 172.20.74.220
Host is up (0.027s latency).
Not shown: 999 filtered ports
PORT    STATE SERVICE
53/tcp open  domain

Nmap done: 1 IP address (1 host up) scanned in 79.04 seconds
netman@netman:~$
```

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**]

Answer:

```
netman@netman:~$ sudo nmap -sP -PI -PT 10.0.0.131/24
```

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

Answer:

Different state options for a NMAP port scan:

Open state:

In open state, application is being accepted by the following connections: TCP, UDP. Attackers usually attack on the open ports, whereas network administrator protect them with the certain firewalls.

Closed state:

In the closed port, no application is listening unlike the open port. It is helpful to show that the host is up on an IP address. It is also a part of the Operating System Detection.

Filtered state:

In the filtered state, Network Management Application Protocol cannot decide about the port whether it is open or closed. This is due to the packet filtering stops its probe from reaching the specific port.

[Reference: https://nmap.org/book/man-port-scanning-basics.html]

2. Provide screenshots of the Operating Systems running on each of these machines [**2.5 points**]

Answer: Below is the screenshot of Operating System running:

```
netman@netman:~$ sudo nmap -V -Pn -O 10.201.23.110
[sudo] password for netman:

Nmap version 7.60 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.3.3 openssl-1.1.0g nmap-libssh2-1.8.0 libz-1.2.8 libpcre-8.39 libpcap-1.8.1 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
netman@netman:~$
```

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**]

Answer: I have scanned for two times, but I didn't find any difference.

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

Answer:

First Scan

```
home > netman > Desktop > Port_scan.py > ...
#!/usr/bin/env python3
#Objective 4: Port Scan Using NMAP

import subprocess as sp

IP_address_scan = "nmap -sP 10.201.23.110/24"
scanning_port = sp.getoutput(IP_address_scan)
print (scanning_port)
with open ('target_file_1','w') as file_1:
    file_1.write(scanning_port)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

netman@netman:~$ /usr/bin/python3 /home/netman/Desktop/Port_scan.py

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 23:14 MST
Nmap scan report for 10.201.23.255
Host is up (0.0023s latency).
Nmap done: 256 IP addresses (1 host up) scanned in 28.88 seconds
netman@netman:~$
```

Second Scan:

```
home > netman > Desktop > nmap > Port_scan.py > ...
#!/usr/bin/env python3
#Objective 4: Port Scan Using NMAP

import subprocess as sp

IP_address_scan = "nmap -sP 10.201.23.110/24"
scanning_port = sp.getoutput(IP_address_scan)
print (scanning_port)
with open ('target_file_1','w') as file_1:
    file_1.write(scanning_port)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

netman@netman:~$ /usr/bin/python3 /home/netman/Desktop/Port_scan.py

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 23:25 MST
Nmap scan report for 10.201.23.255
Host is up (0.0029s latency).
Nmap done: 256 IP addresses (1 host up) scanned in 47.82 seconds
netman@netman:~$
```

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

Answer: NMAP stands for Network Mapper which is open-source application software. This is used for vulnerability scanning and for the discovery of the network. The one thing which is useful as a network administrator, NMAP is used to recognize the machines are running on the network, discovering the hosts that are accessible. Also, the assistance which NMAP give such as finding the ports and encountering the threats. And the one thing which is detrimental to this is security.

2. **Extra Credit:**

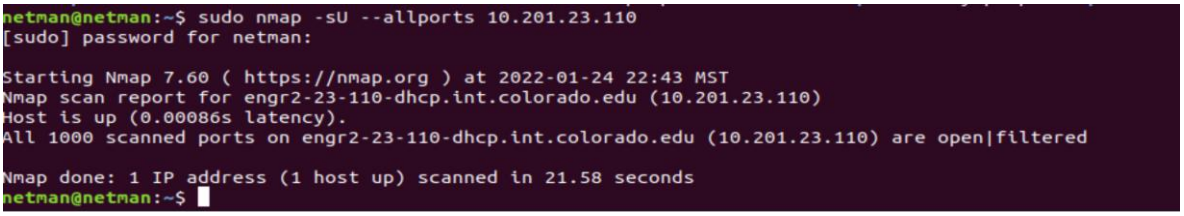
Rogue Web Server (web servers ending with IP addresses .1-.10 are legitimate; outside of that range are rogue)

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)sudo nmap -sU --allports

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

Answer: Below is the screenshot:



```
netman@netman:~$ sudo nmap -sU --allports 10.201.23.110
[sudo] password for netman:

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 22:43 MST
Nmap scan report for engr2-23-110-dhcp.int.colorado.edu (10.201.23.110)
Host is up (0.00086s latency).
All 1000 scanned ports on engr2-23-110-dhcp.int.colorado.edu (10.201.23.110) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 21.58 seconds
netman@netman:~$
```

The command which I have used to run Rogue web servers is:

sudo nmap -sU –allports 10.201.23.110

Report Questions

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

Answer: By using following command, we can able to decoy:
nmap -n -Ddecoy-ip1,decoy-ip2,your-own-ip,decoy-ip3,decoy-ip4 remote-host-ip.
To hide your source IP address using NMAP, we can use a Proxy, VPN, or some other service.

[Reference: https://www.cyberciti.biz/tips/nmap-hide-ipaddress-with-decoy-ideal-scan.html]

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

Answer:

There are many ways NMAP can be used to trick or deceive a firewall as follows:

1. Packet Fragmentation.
2. Spoof source IP address.
3. Decoy scan.

[Reference: https://security.stackexchange.com/questions/121900/how-can-the-nmap-tool-be-used-to-evade-a-firewall-ids]

Total Score = _____/ 167 [+17 Bonus]