Network Management and Automation

Lab 1
Network management using SNMP and NMAP

University of Colorado Boulder
Network Engineering Program

Professor Levi Perigo, Ph.D.

How does a switch work? What information does it learns/stores? (internal process). What is a VLAN and its purpose of existence?   What is STP and its purpose? How STP works? Different types of MAC addresses? (edited)

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 )
```

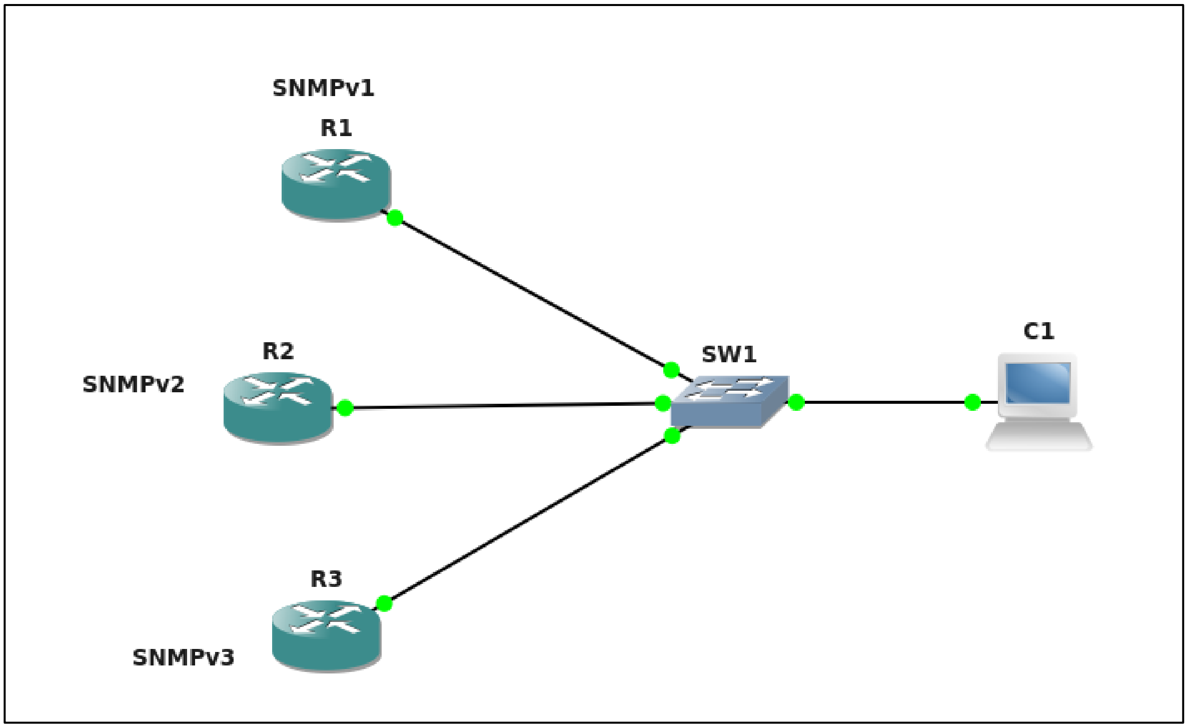
*NOTE: The IP address used within the terminal is the IP address of the Cisco router. In this example the Cisco router has the IP address of 198.51.100.3.

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

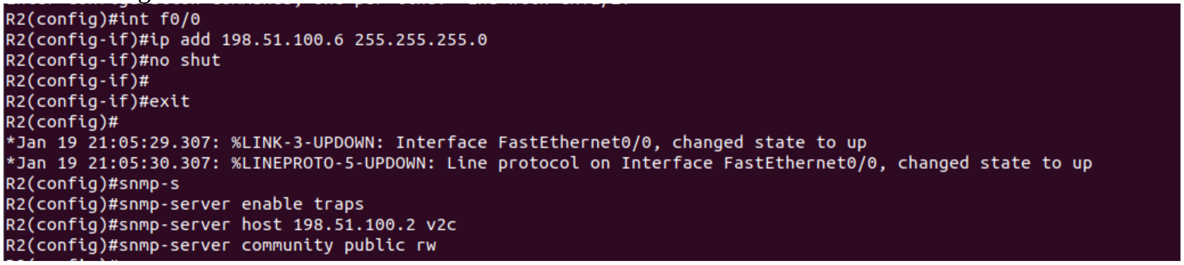
Create the topology in GNS3 as shown below and assign management IPs (198.51.100.0/24 subnet) to them on fa0/0. Configure the nodes for different versions of SNMP & enable traps.

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



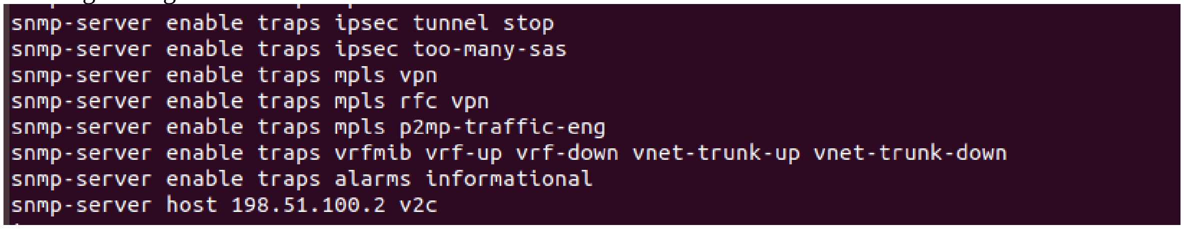
1. How did you configure SNMPv2 and v3 on routers R2 and R3? Provide running configuration screenshots (only portions relevant to SNMP). [**10 points**]

SNMP configuration on R2:



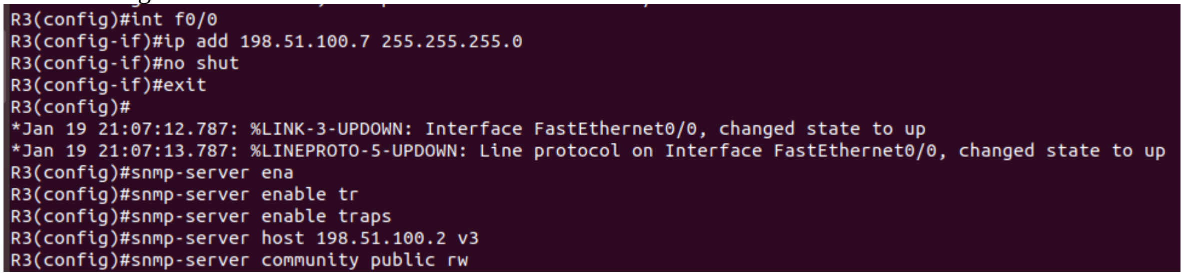
```
R2(config)#int f0/0
R2(config-if)#ip add 198.51.100.6 255.255.255.0
R2(config-if)#no shut
R2(config-if)#
R2(config-if)#exit
R2(config)#
*Jan 19 21:05:29.307: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan 19 21:05:30.307: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config)#snmp-s
R2(config)#snmp-server enable traps
R2(config)#snmp-server host 198.51.100.2 v2c
R2(config)#snmp-server community public rw
```

Running Config on R2:

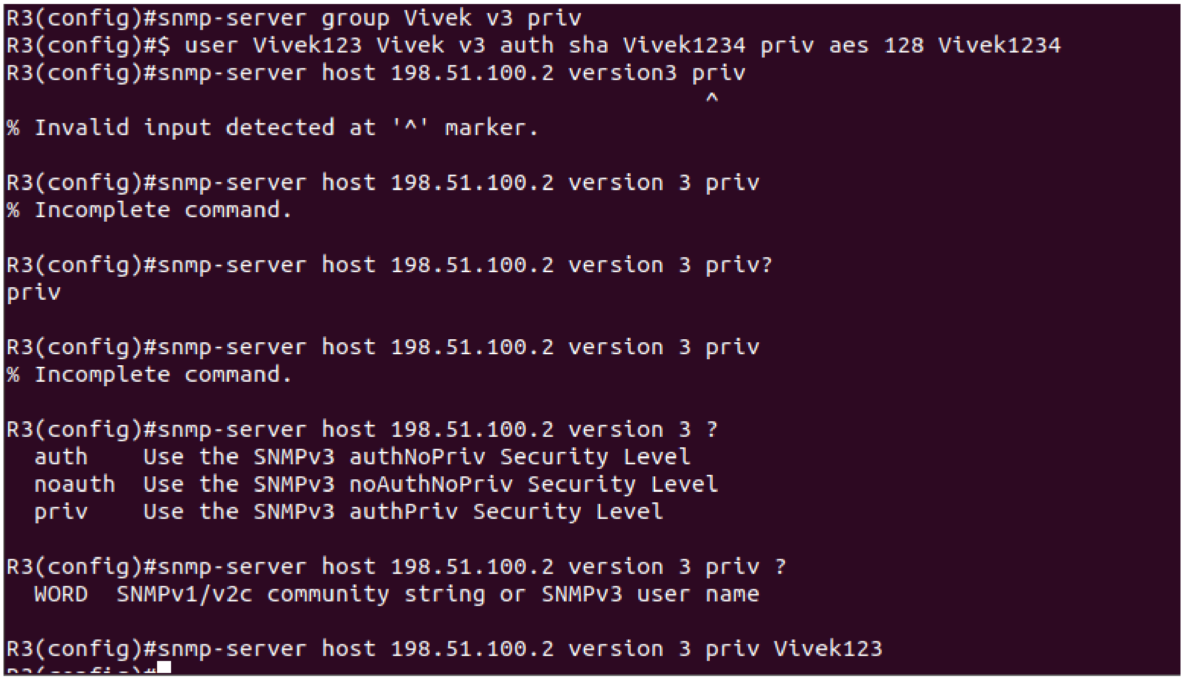


```
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps mpls vpn
snmp-server enable traps mpls rfc vpn
snmp-server enable traps mpls p2mp-traffic-eng
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
snmp-server enable traps alarms informational
snmp-server host 198.51.100.2 v2c
```

SNMP configuration on R3:

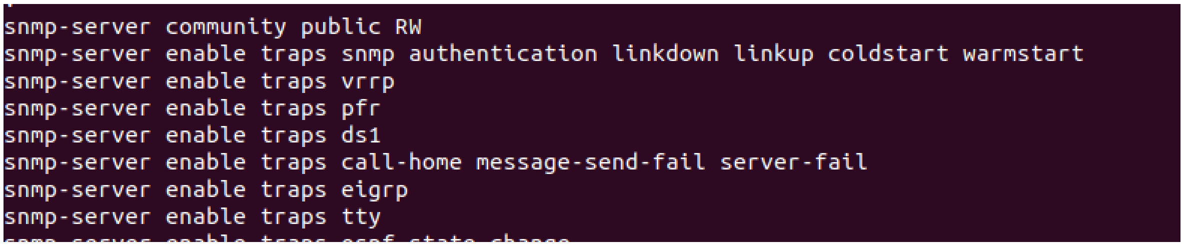


```
R3(config)#int f0/0
R3(config-if)#ip add 198.51.100.7 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Jan 19 21:07:12.787: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan 19 21:07:13.787: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config)#snmp-server ena
R3(config)#snmp-server enable tr
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)#snmp-server group Vivek v3 priv
R3(config)#$ user Vivek123 Vivek v3 auth sha Vivek1234 priv aes 128 Vivek1234
R3(config)#snmp-server host 198.51.100.2 version3 priv
                                                 ^
% Invalid input detected at '^' marker.

R3(config)#snmp-server host 198.51.100.2 version 3 priv
% Incomplete command.

R3(config)#snmp-server host 198.51.100.2 version 3 priv?
priv

R3(config)#snmp-server host 198.51.100.2 version 3 priv
% Incomplete command.

R3(config)#snmp-server host 198.51.100.2 version 3 ?
  auth    Use the SNMPv3 authNoPriv Security Level
  noauth  Use the SNMPv3 noAuthNoPriv Security Level
  priv    Use the SNMPv3 authPriv Security Level

R3(config)#snmp-server host 198.51.100.2 version 3 priv ?
  WORD  SNMPv1/v2c community string or SNMPv3 user name

R3(config)#snmp-server host 198.51.100.2 version 3 priv Vivek123
```

Running Config on R3:



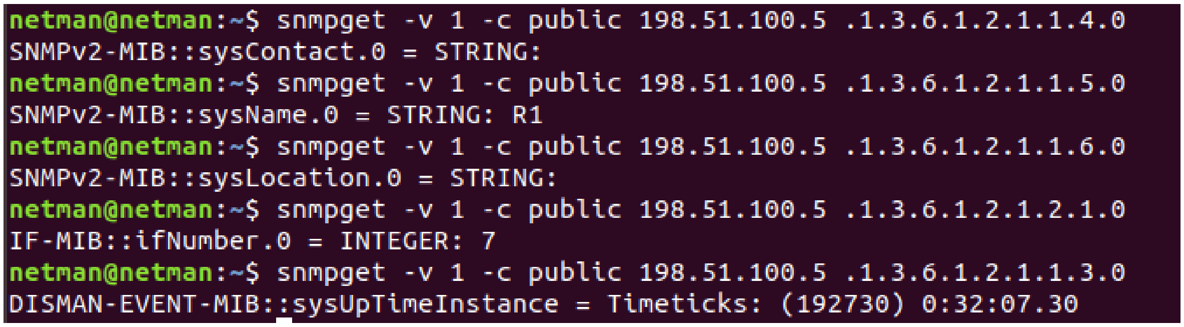
```
snmp-server community public RW
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps pfr
snmp-server enable traps ds1
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps eigrp
snmp-server enable traps tty
snmp-server enable traps ospf state-change
```

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

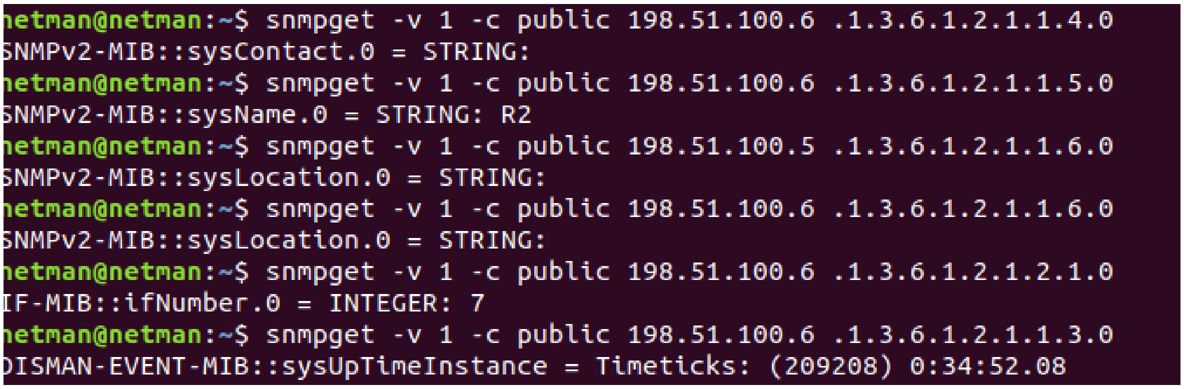
1. Enter the above SNMPGET commands for the OIDs mentioned for SNMP v1, v2, and v3. Paste relevant screenshots. [**10 points**]

   On R1:



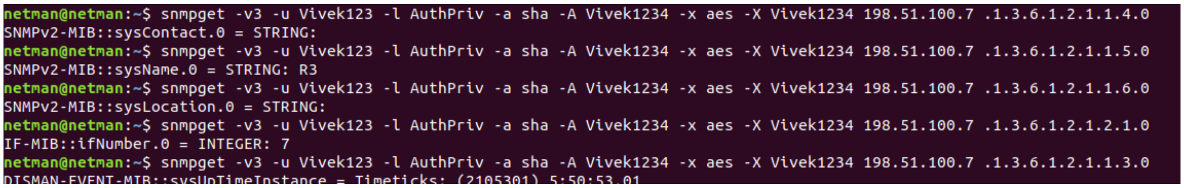
```
netman@netman:~$ snmpget -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING:
netman@netman:~$ snmpget -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: R1
netman@netman:~$ snmpget -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING:
netman@netman:~$ snmpget -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 7
netman@netman:~$ snmpget -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (192730) 0:32:07.30
```

   On R2:



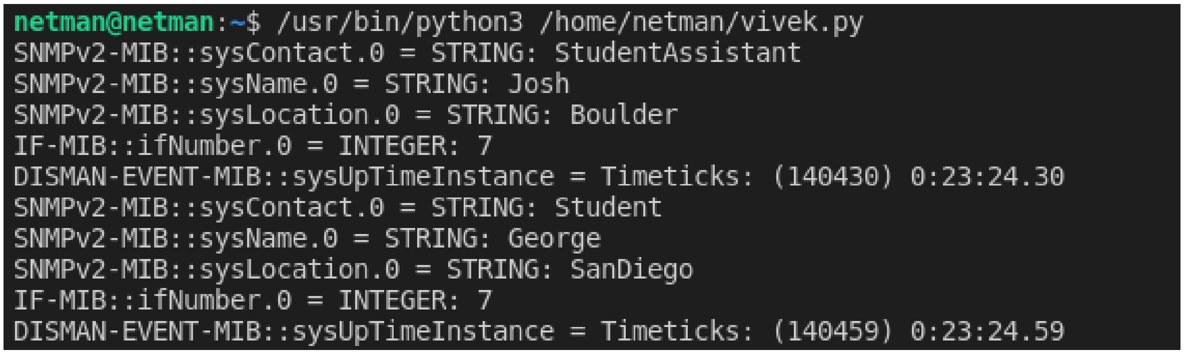
```
netman@netman:~$ snmpget -v 1 -c public 198.51.100.6 .1.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING:
netman@netman:~$ snmpget -v 1 -c public 198.51.100.6 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: R2
netman@netman:~$ snmpget -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING:
netman@netman:~$ snmpget -v 1 -c public 198.51.100.6 .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING:
netman@netman:~$ snmpget -v 1 -c public 198.51.100.6 .1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 7
netman@netman:~$ snmpget -v 1 -c public 198.51.100.6 .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (209208) 0:34:52.08
```

   On R3:



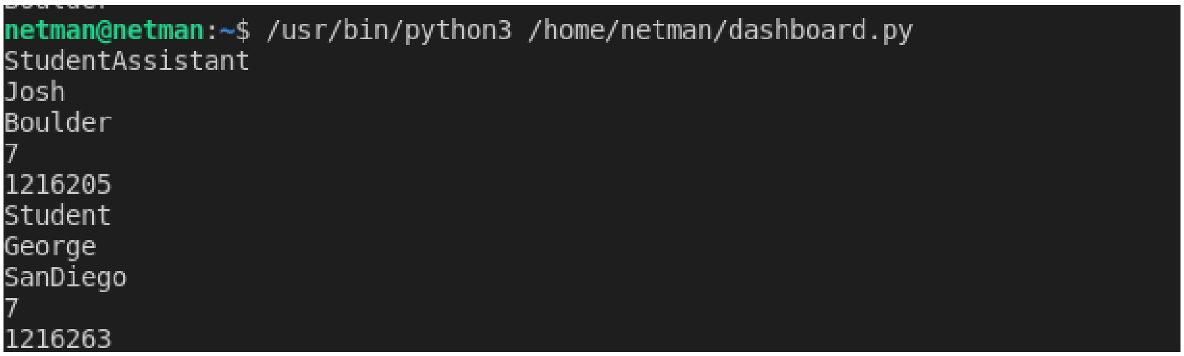
```
netman@netman:~$ snmpget -v3 -u Vivek123 -l AuthPriv -a sha -A Vivek1234 -x aes -X Vivek1234 198.51.100.7 .1.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING:
netman@netman:~$ snmpget -v3 -u Vivek123 -l AuthPriv -a sha -A Vivek1234 -x aes -X Vivek1234 198.51.100.7 .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: R3
netman@netman:~$ snmpget -v3 -u Vivek123 -l AuthPriv -a sha -A Vivek1234 -x aes -X Vivek1234 198.51.100.7 .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING:
netman@netman:~$ snmpget -v3 -u Vivek123 -l AuthPriv -a sha -A Vivek1234 -x aes -X Vivek1234 198.51.100.7 .1.3.6.1.2.1.2.1.0
IF-MIB::ifNumber.0 = INTEGER: 7
netman@netman:~$ snmpget -v3 -u Vivek123 -l AuthPriv -a sha -A Vivek1234 -x aes -X Vivek1234 198.51.100.7 .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (2105301) 5:50:53.01
```

2. Create a dashboard to display the output from those commands using UNIX/Python. Paste relevant screenshots. [**15 points**]
   I did on R1 and R2 as it was mentioned on any two.



```
netman@netman:~$ /usr/bin/python3 /home/netman/vivek.py
SNMPv2-MIB::sysContact.0 = STRING: StudentAssistant
SNMPv2-MIB::sysName.0 = STRING: Josh
SNMPv2-MIB::sysLocation.0 = STRING: Boulder
IF-MIB::ifNumber.0 = INTEGER: 7
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (140430) 0:23:24.30
SNMPv2-MIB::sysContact.0 = STRING: Student
SNMPv2-MIB::sysName.0 = STRING: George
SNMPv2-MIB::sysLocation.0 = STRING: SanDiego
IF-MIB::ifNumber.0 = INTEGER: 7
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (140459) 0:23:24.59
```

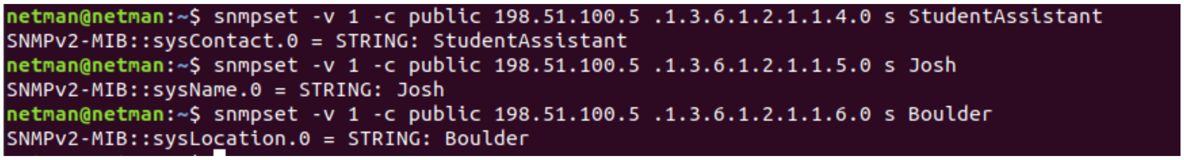
Dashboard for displaying the different OID’s.



```
netman@netman:~$ /usr/bin/python3 /home/netman/dashboard.py
StudentAssistant
Josh
Boulder
7
1216205
Student
George
SanDiego
7
1216263
```

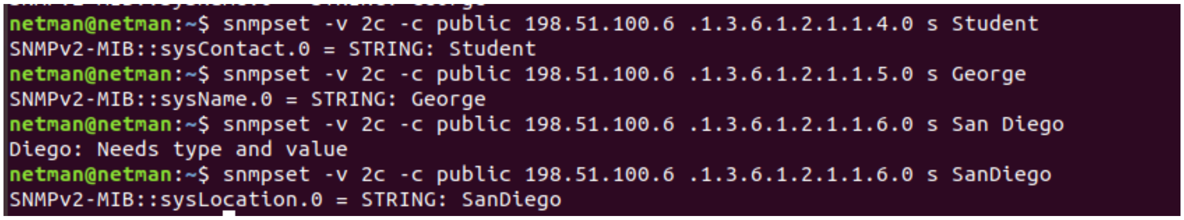
3. Use SNMPSET commands to modify Contact, Name, and Location to display varied output for each version: 1 and 2. Paste relevant screenshots. [**10 points**]

Modifictions on R1:



```
netman@netman:~$ snmpset -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.4.0 s StudentAssistant
SNMPv2-MIB::sysContact.0 = STRING: StudentAssistant
netman@netman:~$ snmpset -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.5.0 s Josh
SNMPv2-MIB::sysName.0 = STRING: Josh
netman@netman:~$ snmpset -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.6.0 s Boulder
SNMPv2-MIB::sysLocation.0 = STRING: Boulder
```

Modifications on R2:



```
netman@netman:~$ snmpset -v 2c -c public 198.51.100.6 .1.3.6.1.2.1.1.4.0 s Student
SNMPv2-MIB::sysContact.0 = STRING: Student
netman@netman:~$ snmpset -v 2c -c public 198.51.100.6 .1.3.6.1.2.1.1.5.0 s George
SNMPv2-MIB::sysName.0 = STRING: George
netman@netman:~$ snmpset -v 2c -c public 198.51.100.6 .1.3.6.1.2.1.1.6.0 s San Diego
Diego: Needs type and value
netman@netman:~$ snmpset -v 2c -c public 198.51.100.6 .1.3.6.1.2.1.1.6.0 s SanDiego
SNMPv2-MIB::sysLocation.0 = STRING: SanDiego
```

Sample dashboard to be displayed using UNIX/Python:

SNMP v1
Contact: Student Assistant
Name: Josh
Location: Boulder
Number: 2
Uptime: 0:54:20.47

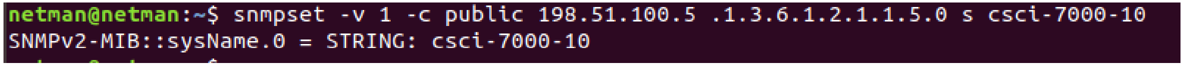
SNMP v2
Contact: Student
Name: George
Location: San Diego
Number: 2
Uptime: 0:67:10.57

SNMP v3 (any of the 2)
Contact: Professor
Name: Kelly
Location: Dallas
Number: 2
Uptime: 1:24:20.47

Objective 3: SNMPSET Commands

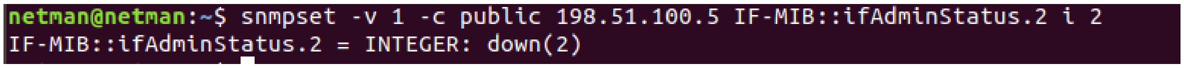
NOTE: Must use SNMPSET commands to perform the below tasks on Router 1 in GNS3:

1. Change the hostname to "**csci-7000-10**" (provide a screenshot) [**10 points**]



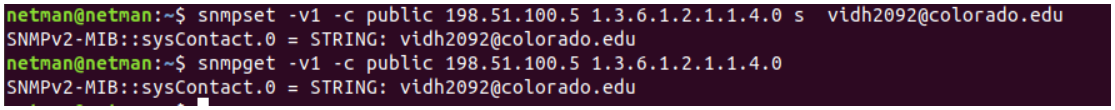
```
netman@netman:~$ snmpset -v 1 -c public 198.51.100.5 .1.3.6.1.2.1.1.5.0 s csci-7000-10
SNMPv2-MIB::sysName.0 = STRING: csci-7000-10
```

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**]



```
netman@netman:~$ snmpset -v 1 -c public 198.51.100.5 IF-MIB::ifAdminStatus.2 i 2
IF-MIB::ifAdminStatus.2 = INTEGER: down(2)
```

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



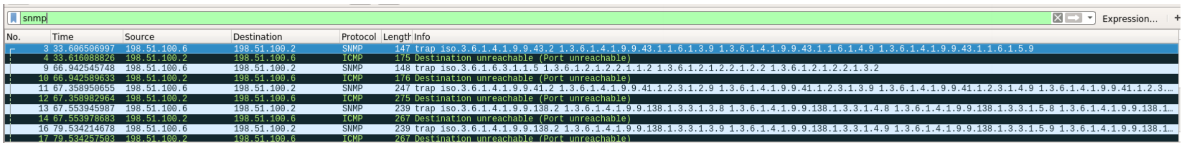
```
netman@netman:~$ snmpset -v1 -c public 198.51.100.5 1.3.6.1.2.1.1.4.0 s  vidh2092@colorado.edu
SNMPv2-MIB::sysContact.0 = STRING: vidh2092@colorado.edu
netman@netman:~$ snmpget -v1 -c public 198.51.100.5 1.3.6.1.2.1.1.4.0
SNMPv2-MIB::sysContact.0 = STRING: vidh2092@colorado.edu
```

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.

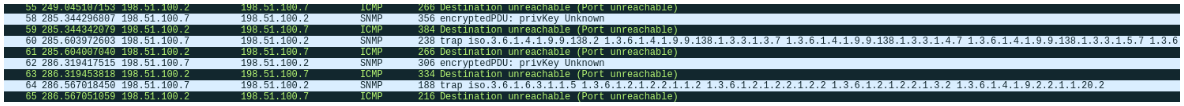
2. Shutdown the interfaces on R2 and R3, and bring them up again. Do you observe different trap messages being exchanged between the SNMP agent and the manager (VM) in the packet capture? Provide relevant screenshots. [**10 points**]

On R2:



```
snmp
No.  Time          Source         Destination    Protocol Length Info
   3 33.606506997  198.51.100.6   198.51.100.2   SNMP     147 trap iso.3.6.1.4.1.9.9.43.2 1.3.6.1.4.1.9.9.43.1.1.6.1.3.9 1.3.6.1.4.1.9.9.43.1.1.6.1.4.9 1.3.6.1.4.1.9.9.43.1.1.6.1.5.9
   4 33.616088826  198.51.100.2   198.51.100.6   ICMP     175 Destination unreachable (Port unreachable)
   9 66.942545748  198.51.100.6   198.51.100.2   SNMP     148 trap iso.3.6.1.6.3.1.1.5 1.3.6.1.2.1.2.2.1.1.2 1.3.6.1.2.1.2.2.1.2.2 1.3.6.1.2.1.2.2.1.3.2
  10 66.942589633  198.51.100.2   198.51.100.6   ICMP     176 Destination unreachable (Port unreachable)
  11 67.358950655  198.51.100.6   198.51.100.2   SNMP     247 trap iso.3.6.1.4.1.9.9.41.2 1.3.6.1.4.1.9.9.41.1.2.3.1.2.9 1.3.6.1.4.1.9.9.41.1.2.3.1.3.9 1.3.6.1.4.1.9.9.41.1.2.3.1.4.9 1.3.6.1.4.1.9.9.41.1.2.3…
  12 67.358982964  198.51.100.2   198.51.100.6   ICMP     275 Destination unreachable (Port unreachable)
  13 67.553945987  198.51.100.6   198.51.100.2   SNMP     239 trap iso.3.6.1.4.1.9.9.138.2 1.3.6.1.4.1.9.9.138.1.3.3.1.3.8 1.3.6.1.4.1.9.9.138.1.3.3.1.4.8 1.3.6.1.4.1.9.9.138.1.3.3.1.5.8 1.3.6.1.4.1.9.9.138.1…
  14 67.553978683  198.51.100.2   198.51.100.6   ICMP     267 Destination unreachable (Port unreachable)
  16 79.534214678  198.51.100.6   198.51.100.2   SNMP     239 trap iso.3.6.1.4.1.9.9.138.2 1.3.6.1.4.1.9.9.138.1.3.3.1.3.9 1.3.6.1.4.1.9.9.138.1.3.3.1.4.9 1.3.6.1.4.1.9.9.138.1.3.3.1.5.9 1.3.6.1.4.1.9.9.138.1…
  17 79.534257503  198.51.100.2   198.51.100.6   ICMP     267 Destination unreachable (Port unreachable)
```

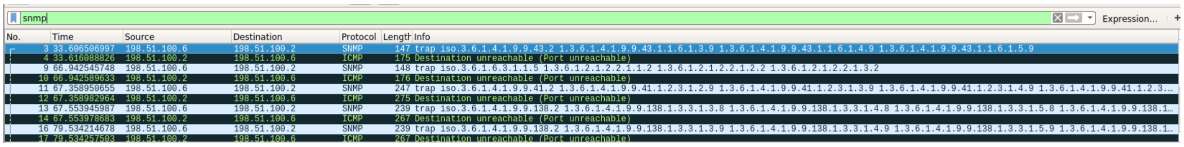
On R3:



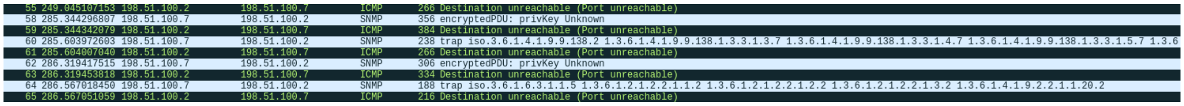
```
55 249.045107153 198.51.100.2   198.51.100.7   ICMP   266 Destination unreachable (Port unreachable)
58 285.344296807 198.51.100.7   198.51.100.2   SNMP   356 encryptedPDU: privKey Unknown
59 285.344342079 198.51.100.2   198.51.100.7   ICMP   384 Destination unreachable (Port unreachable)
60 285.603972603 198.51.100.7   198.51.100.2   SNMP   238 trap iso.3.6.1.4.1.9.9.138.2 1.3.6.1.4.1.9.9.138.1.3.3.1.3.7 1.3.6.1.4.1.9.9.138.1.3.3.1.4.7 1.3.6.1.4.1.9.9.138.1.3.3.1.5.7 1.3.6
61 285.604007040 198.51.100.2   198.51.100.7   ICMP   266 Destination unreachable (Port unreachable)
62 286.319417515 198.51.100.7   198.51.100.2   SNMP   306 encryptedPDU: privKey Unknown
63 286.319453818 198.51.100.2   198.51.100.7   ICMP   334 Destination unreachable (Port unreachable)
64 286.567018450 198.51.100.7   198.51.100.2   SNMP   188 trap iso.3.6.1.6.3.1.1.5 1.3.6.1.2.1.2.2.1.1.2 1.3.6.1.2.1.2.2.1.2.2 1.3.6.1.2.1.2.2.1.3.2 1.3.6.1.4.1.9.2.2.1.1.20.2
65 286.567051059 198.51.100.2   198.51.100.7   ICMP   216 Destination unreachable (Port unreachable)
```

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**]

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**]
The main difference between the trap messages for SNMPv2 and SNMPv3 is the trap messages being captured for each one. SNMPv3 is more secure as compared to SNMPv2. There is no security for v2 but there is an additional encryption in v3. As seen in below ss that v3 has encrypted PDU privkey unknown which is the main difference between the two.



```
snmp
No.  Time          Source         Destination    Protocol Length Info
   3 33.606506997  198.51.100.6   198.51.100.2   SNMP     147 trap iso.3.6.1.4.1.9.9.43.2 1.3.6.1.4.1.9.9.43.1.1.6.1.3.9 1.3.6.1.4.1.9.9.43.1.1.6.1.4.9 1.3.6.1.4.1.9.9.43.1.1.6.1.5.9
   4 33.616088826  198.51.100.2   198.51.100.6   ICMP     175 Destination unreachable (Port unreachable)
   9 66.942545748  198.51.100.6   198.51.100.2   SNMP     148 trap iso.3.6.1.6.3.1.1.5 1.3.6.1.2.1.2.2.1.1.2 1.3.6.1.2.1.2.2.1.2.2 1.3.6.1.2.1.2.2.1.3.2
  10 66.942589633  198.51.100.2   198.51.100.6   ICMP     176 Destination unreachable (Port unreachable)
  11 67.358950655  198.51.100.6   198.51.100.2   SNMP     247 trap iso.3.6.1.4.1.9.9.41.2 1.3.6.1.4.1.9.9.41.1.2.3.1.2.9 1.3.6.1.4.1.9.9.41.1.2.3.1.3.9 1.3.6.1.4.1.9.9.41.1.2.3.1.4.9 1.3.6.1.4.1.9.9.41.1.2.3…
  12 67.358982964  198.51.100.2   198.51.100.6   ICMP     275 Destination unreachable (Port unreachable)
  13 67.553945987  198.51.100.6   198.51.100.2   SNMP     239 trap iso.3.6.1.4.1.9.9.138.2 1.3.6.1.4.1.9.9.138.1.3.3.1.3.8 1.3.6.1.4.1.9.9.138.1.3.3.1.4.8 1.3.6.1.4.1.9.9.138.1.3.3.1.5.8 1.3.6.1.4.1.9.9.138.1…
  14 67.553978683  198.51.100.2   198.51.100.6   ICMP     267 Destination unreachable (Port unreachable)
  16 79.534214678  198.51.100.6   198.51.100.2   SNMP     239 trap iso.3.6.1.4.1.9.9.138.2 1.3.6.1.4.1.9.9.138.1.3.3.1.3.9 1.3.6.1.4.1.9.9.138.1.3.3.1.4.9 1.3.6.1.4.1.9.9.138.1.3.3.1.5.9 1.3.6.1.4.1.9.9.138.1…
  17 79.534257503  198.51.100.2   198.51.100.6   ICMP     267 Destination unreachable (Port unreachable)
```



```
55 249.045107153 198.51.100.2    198.51.100.7    ICMP    266 Destination unreachable (Port unreachable)
58 285.344296807 198.51.100.7    198.51.100.2    SNMP    356 encryptedPDU: privKey Unknown
59 285.344342079 198.51.100.2    198.51.100.7    ICMP    384 Destination unreachable (Port unreachable)
60 285.603972603 198.51.100.7    198.51.100.2    SNMP    238 trap iso.3.6.1.4.1.9.9.138.2 1.3.6.1.4.1.9.9.138.1.3.3.1.3.7 1.3.6.1.4.1.9.9.138.1.3.3.1.4.7 1.3.6.1.4.1.9.9.138.1.3.3.1.5.7 1.3.6
61 285.604007040 198.51.100.2    198.51.100.7    ICMP    266 Destination unreachable (Port unreachable)
62 286.319417515 198.51.100.7    198.51.100.2    SNMP    306 encryptedPDU: privKey Unknown
63 286.319453818 198.51.100.2    198.51.100.7    ICMP    334 Destination unreachable (Port unreachable)
64 286.567018450 198.51.100.7    198.51.100.2    SNMP    188 trap iso.3.6.1.6.3.1.1.5 1.3.6.1.2.1.2.2.1.1.2 1.3.6.1.2.1.2.2.1.2.2 1.3.6.1.2.1.2.2.1.3.2 1.3.6.1.4.1.9.2.2.1.1.20.2
65 286.567051059 198.51.100.2    198.51.100.7    ICMP    216 Destination unreachable (Port unreachable)
```

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?
   Yes, using a management subnet for SNMP gives an additional security. It allows only the management IP's to enter the network and blocking all the other IP addresses. Therefore it makes the network more secure.
2. Why is a switch used in the network design in GNS3?
   Switch allows devices to communicate on the same network. It allows resource sharing. Switch filters and forwards the packets. It is a hardware device that filters and forwards packets. So, to allow devices to communicate with each other on the same network we have to use switch.
3. Can you use a router instead? Why/why not?
   No, within the network to allow internetwork communication we have to use switch, router is used to get to a different network. It is used to route packets to different network depending upon the destination IP address. Also, router is costly as compared to switch so its better to use switch and not router instead of switch.
4. If you used a router, what would need to change (if anything)
   We will have to connect the LAN port on the router to the LAN port on the switch. We cannot use WAN port on router.
5. What command has to be entered on the router, to disable configuration changes to be made through SNMP?
   SNMP host disable command disables configuration changes to be made through SNMP.

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

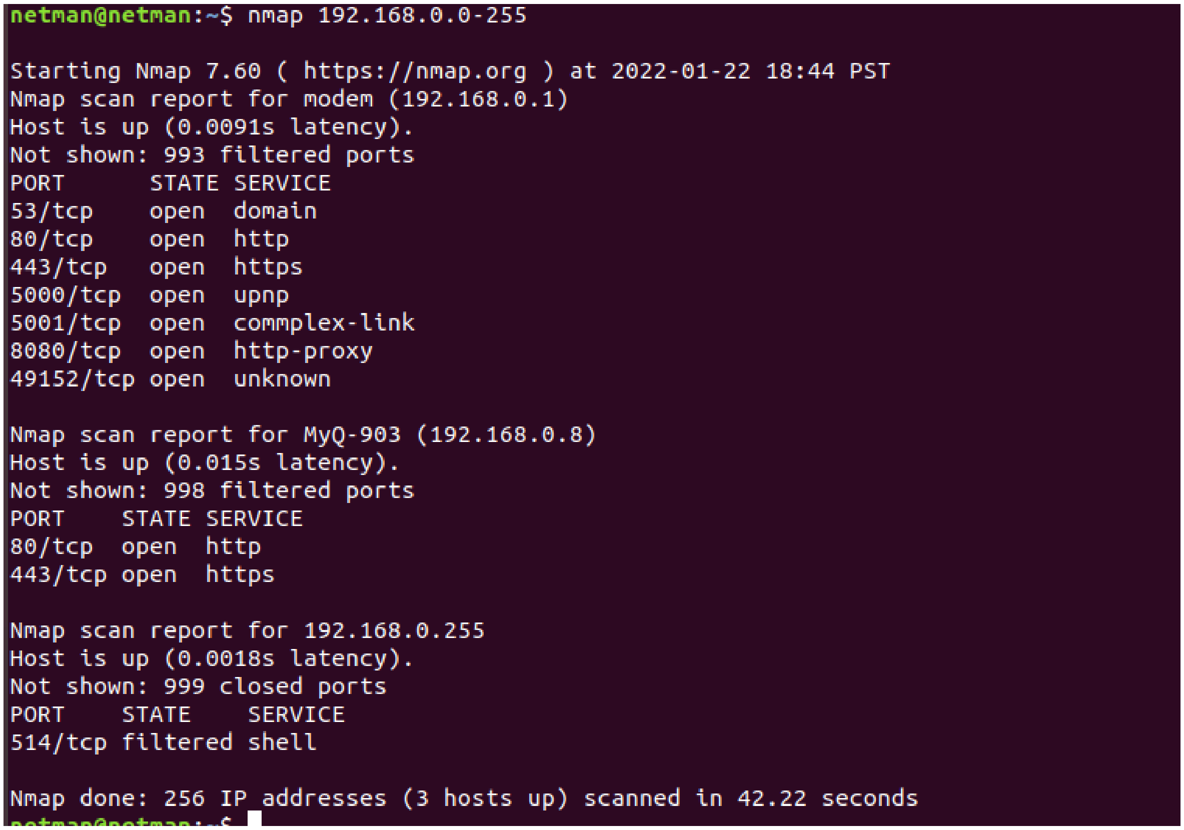
Follow the instructions from the Nmap website for your operating system:

https://nmap.org/

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

Objective 2: Ping Sweeps and Port Scans

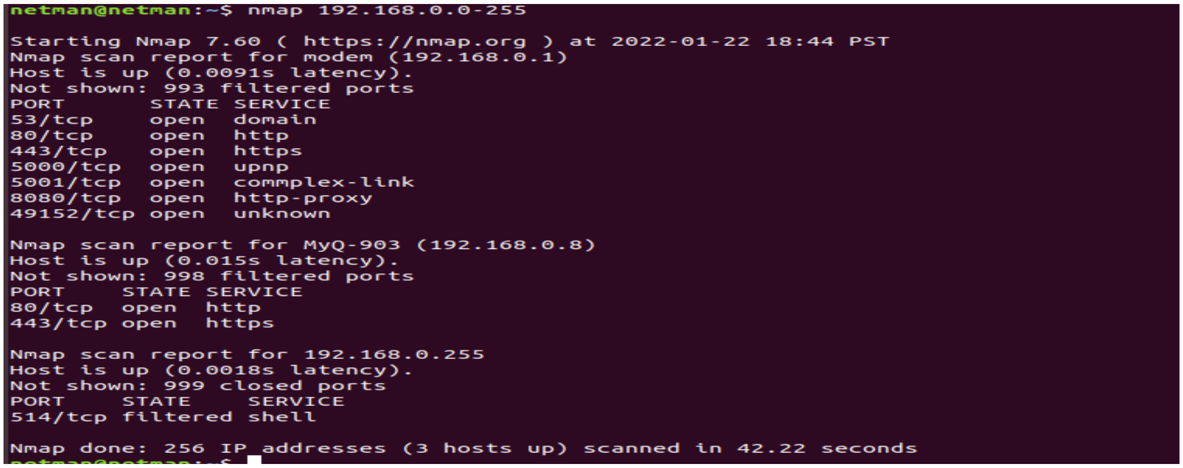
1. Perform a ping sweep for the following network (Note: this only works from CU network or VPN; if unavailable use your home/private network):
   172.20.74.0/24
   a. Provide a screenshot showing the command and the results [**5 points**]
      Since the network was unavailable I did it for my private network at home.



```
netman@netman:~$ nmap 192.168.0.0-255

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-22 18:44 PST
Nmap scan report for modem (192.168.0.1)
Host is up (0.0091s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
5000/tcp  open  upnp
5001/tcp  open  commplex-link
8080/tcp  open  http-proxy
49152/tcp open  unknown

Nmap scan report for MyQ-903 (192.168.0.8)
Host is up (0.015s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.255
Host is up (0.0018s latency).
Not shown: 999 closed ports
PORT    STATE    SERVICE
514/tcp filtered shell

Nmap done: 256 IP addresses (3 hosts up) scanned in 42.22 seconds
netman@netman:~$
```

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

3 devices responded to the ping sweep. I found out this by using the information provided after running the above command.



```
netman@netman:~$ nmap 192.168.0.0-255

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-22 18:44 PST
Nmap scan report for modem (192.168.0.1)
Host is up (0.0091s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
5000/tcp  open  upnp
5001/tcp  open  commplex-link
8080/tcp  open  http-proxy
49152/tcp open  unknown

Nmap scan report for MyQ-903 (192.168.0.8)
Host is up (0.015s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.255
Host is up (0.0018s latency).
Not shown: 999 closed ports
PORT    STATE    SERVICE
514/tcp filtered shell

Nmap done: 256 IP addresses (3 hosts up) scanned in 42.22 seconds
netman@netman:~$
```

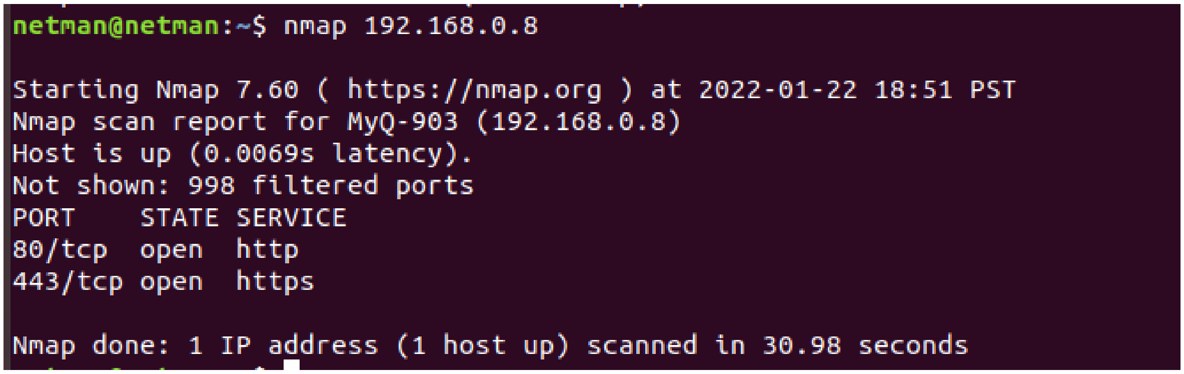
2. Choose a host that replied from the ping sweep; now perform a full scan on that host

   a. Which well-known ports were open on this machine?

      Provide the screenshot. [**2.5 points**]

As seen below in the screenshot we can see port 80 and port 443

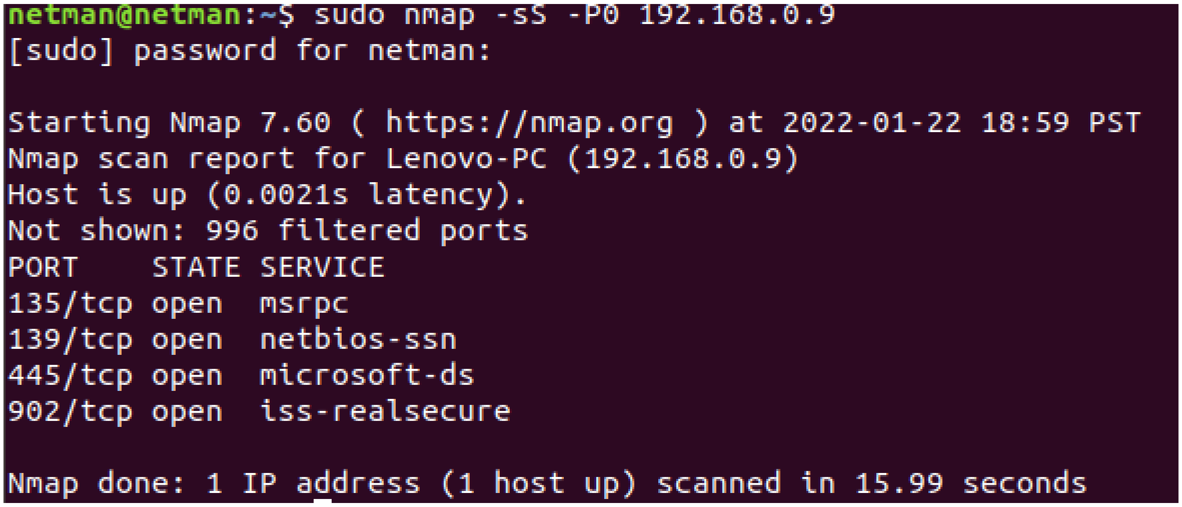
are open.



```
netman@netman:~$ nmap 192.168.0.8

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-22 18:51 PST
Nmap scan report for MyQ-903 (192.168.0.8)
Host is up (0.0069s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 30.98 seconds
```

c. Provide the command you would use to perform a “stealth” scan.

[**2.5 points**]

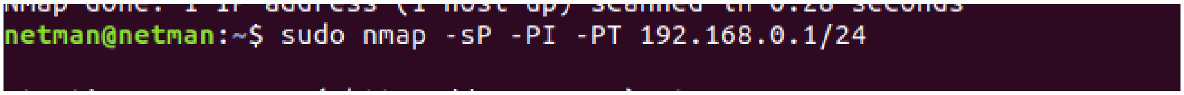


```
netman@netman:~$ sudo nmap -sS -P0 192.168.0.9
[sudo] password for netman:

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-22 18:59 PST
Nmap scan report for Lenovo-PC (192.168.0.9)
Host is up (0.0021s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure

Nmap done: 1 IP address (1 host up) scanned in 15.99 seconds
```

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**]



```
netman@netman:~$ sudo nmap -sP -PI -PT 192.168.0.1/24
```

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

   The different NMAP ports states are open, filtered, closed, unfiltered, open-filtered, closed-filtered.

a. Open- The port is open and accepting TCP connections.

b. filtered-It is unable to find whether the port is open because packet filtering prevents from reaching the port.

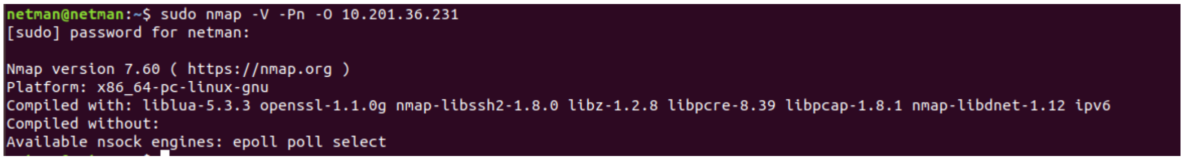
c. Closed-The port is closed and no application is listening on it.

d. Unfiltered- The port is accessible but nmap cannot find whether it is open or not.

e. Open-filtered- It places the port in this state when it doesn’t know if the port is open or filtered.

f. Closed-filtered- It places the port in this state when it doesn’t know if the port is closed or filtered.

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



```
netman@netman:~$ sudo nmap -V -Pn -O 10.201.36.231
[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
```

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

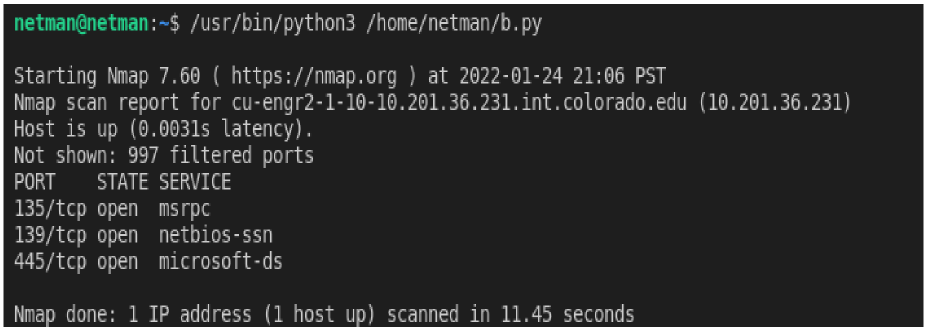
   c. Using **Bash or Python**, record the IP addresses into a **text/CSV**

   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**] Yes, I noted the difference between the two files after running it after 10 min. The difference was I found an additional IP in the later ping sweep.

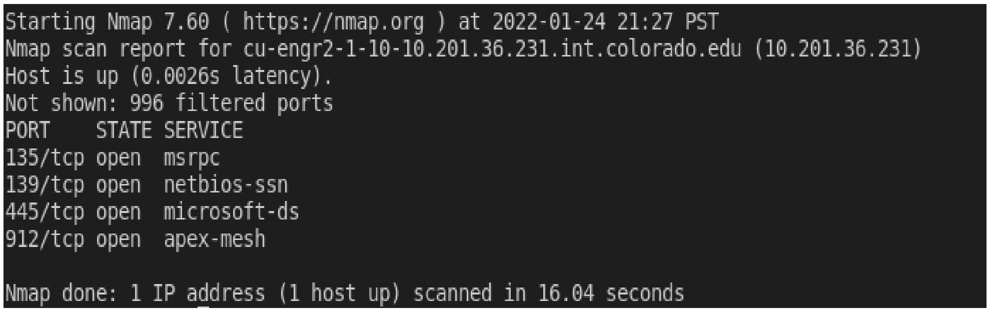
ii. Submit the scripts, files, procedures or screenshots of how you accomplished this [**10 points**]



```
netman@netman:~$ /usr/bin/python3 /home/netman/b.py

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 21:06 PST
Nmap scan report for cu-engr2-1-10-10.201.36.231.int.colorado.edu (10.201.36.231)
Host is up (0.0031s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 11.45 seconds
```

After 10min:

As seen from the below screenshot we can see there is an additional port open which is 912/tcp apex-mesh.



```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 21:27 PST
Nmap scan report for cu-engr2-1-10-10.201.36.231.int.colorado.edu (10.201.36.231)
Host is up (0.0026s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
912/tcp open  apex-mesh

Nmap done: 1 IP address (1 host up) scanned in 16.04 seconds
```

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

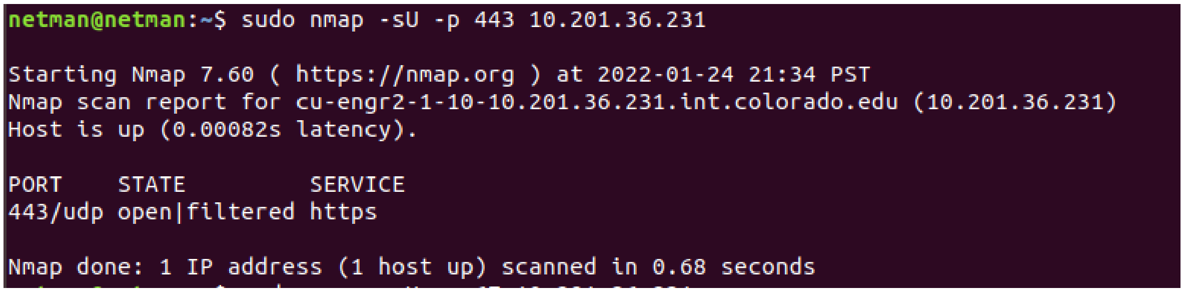
The biggest drawback of using NMAP is security. Any malicious user can do port scanning and find out the active devices in the network. Bad guy can also impersonate the IP/MAC address by some spoofing and act as an original user possessing threat to the security. Useful thing is we get to know the ports active, hosts up, and can be useful for troubleshooting.

2. **Extra Credit:**

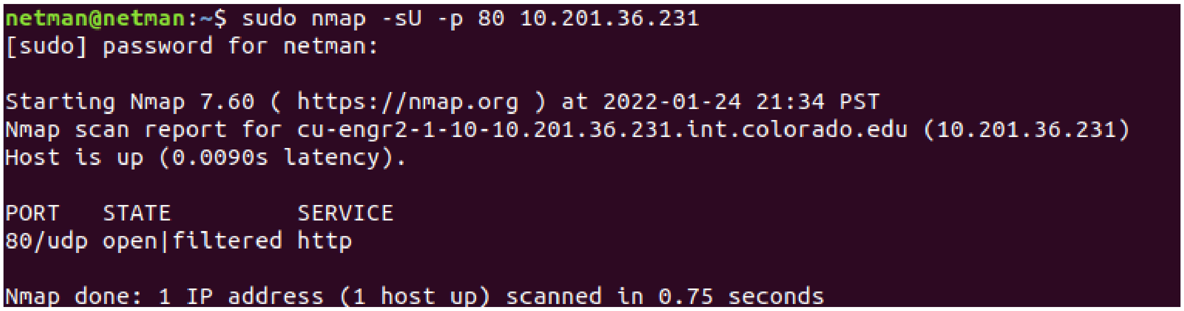
Rogue Web Server (web servers ending with IP addresses .1-.10 are legitimate;

outside of that range are rogue)

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



```
netman@netman:~$ sudo nmap -sU -p 443 10.201.36.231

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 21:34 PST
Nmap scan report for cu-engr2-1-10-10.201.36.231.int.colorado.edu (10.201.36.231)
Host is up (0.00082s latency).

PORT    STATE         SERVICE
443/udp open|filtered https

Nmap done: 1 IP address (1 host up) scanned in 0.68 seconds
```

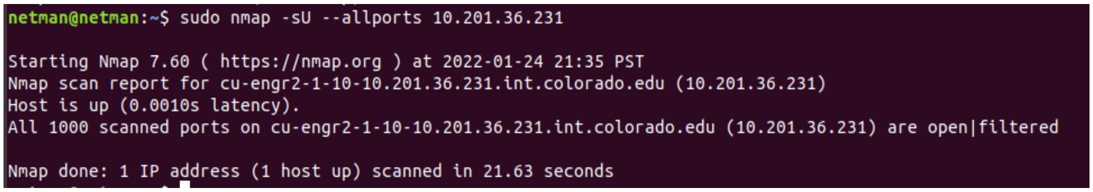


```
netman@netman:~$ sudo nmap -sU -p 80 10.201.36.231
[sudo] password for netman:

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 21:34 PST
Nmap scan report for cu-engr2-1-10-10.201.36.231.int.colorado.edu (10.201.36.231)
Host is up (0.0090s latency).

PORT   STATE         SERVICE
80/udp open|filtered http

Nmap done: 1 IP address (1 host up) scanned in 0.75 seconds
```

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**]

I did port scanning for all the ports and it showed all the ports are open but filtered so that means there is no any rogue server.



```
netman@netman:~$ sudo nmap -sU --allports 10.201.36.231

Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-24 21:35 PST
Nmap scan report for cu-engr2-1-10-10.201.36.231.int.colorado.edu (10.201.36.231)
Host is up (0.0010s latency).
All 1000 scanned ports on cu-engr2-1-10-10.201.36.231.int.colorado.edu (10.201.36.231) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 21.63 seconds
```

Report Questions

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

Nmap with -D allows to do a decoy scan. This can be done as:

Nmap -P0 -sI 1.1.1.1:1234 192.168.0.10. It uses an decoy with port 1234 on 1.1.1.1 to scan host 192.168.0.10.

-Reference: cybercity.biz

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

1.Packet fragmentation
2. Decoy addresses.
3.Idle port scan.
4. Send bad checksums.

Reference – pentestlab.blog

Total Score = ___________________/ 167 [+17 Bonus]