Arquitetura e Gestão de Redes

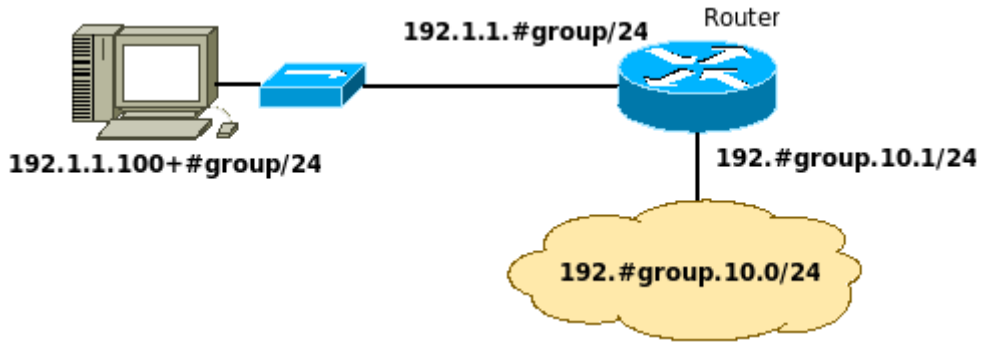
Laboratory Guide

Objectives

- Simple Network Management Protocol (SNMP) – Configuration, usage and security

SNMP v2c

1. Configure an Ethernet network according to the following figure.



2. At the router, configure a SNMP community (using the default name "public") with Read-Only permissions:

```
Router(config)# snmp-server community public RO
```

Start a capture with Wireshark. Using the Linux SNMP tools (type "*man snmpwalk*" for more details), retrieve the Router's complete MIB information (starting from .1 base object ID):

```
snmpwalk -v2c -c public 192.1.1.#group .1
```

Analyze the information present in Router's MIB and the captured SNMP packets.

3. Retrieve partial MIB information using a filter string:

```
snmpwalk -v2c -c public 192.1.1.#group <filter>
```

Retrieve the Router's general information:

```
snmpwalk -v2c -c public 192.1.1.#group sysDescr
snmpwalk -v2c -c public 192.1.1.#group .1.3.6.1.2.1.1.1
```

Retrieve the Router's ARP table:

```
snmpwalk -v2c -c public 192.1.1.#group at
snmpwalk -v2c -c public 192.1.1.#group .1.3.6.1.2.1.3
```

Retrieve the Router's IP routing table:

```
snmpwalk -v2c -c public 192.1.1.#group ipRoute
snmpwalk -v2c -c public 192.1.1.#group .1.3.6.1.2.1.4.21
```

Retrieve the Router's interfaces information and identify the interfaces' names and status:

```
snmpwalk -v2c -c public 192.1.1.#group interfaces
snmpwalk -v2c -c public 192.1.1.#group .1.3.6.1.2.1.2
```

Try other filter strings. Also, analyze the contents of the MIB CISCO-RHINO-MIB located at ~/.snmp/mibs.

4. Start a capture with Wireshark and try to obtain a specific MIB entry. For system description:

```
snmpget -v2c -c public 192.1.1.#group SNMPv2-MIB::sysDescr.0
snmpget -v2c -c public 192.1.1.#group .1.3.6.1.2.1.1.1.0
```

Try to obtain other MIB objects.

5. Start a capture with Wireshark and try to change the status of the Ethernet interface connected to 192.#group.10.0/24 using the *snmpset* command (type "*man snmpset*" for more details):

```
snmpset -v2c -c public 192.1.1.#group IF-MIB::ifAdminStatus.2 i 2
```

Create a new community with Read-Write permission

```
Router(config)# snmp-server community myrouter1 RW
```

Retry the above *snmpset* command with the new community, verify at the router the correct change of the interface status and analyze the captured SNMP packets.

6. **For security reasons never use common community names (e.g public, private, etc...).** Therefore, remove the public community defined above and give the RO community another name:

```
Router(config)# no snmp-server community public RO
Router(config)# snmp-server community myrouter0 RO
```

Test the new community:

```
snmpwalk -v2c -c myrouter0 192.1.1.#group sysDescr
```

7. **For security reasons the SNMP access should be restricted to some IP addresses.** Restrict the access to myrouter0 community to your PC (IP address 192.1.1.100+#group):

```
Router(config)# access-list 10 permit 192.1.1.100+#group
Router(config)# snmp-server community myrouter0 RO 10
```

Test the configuration by accessing the MIB from your PC:

```
snmpwalk -v2c -c myrouter0 192.1.1.#group sysDescr
```

8. Redefine the access restrictions to allow a RO access to all computers in the network 192.1.1.0/24 and a RW access just to your PC. Test the configuration.

9. **For security reasons the SNMP access should be restricted to some MIB objects.** Define a MIB view restriction just to allow the RO access to the system objects:

```
Router(config)# snmp-server view myview system included
Router(config)# snmp-server community myrouter0 view myview RO 10
```

Test the configuration by trying to access the Router's interfaces information:

```
snmpwalk -v2c -c myrouter0 192.1.1.#group interfaces
```

Redefine the SNMP view to allow the access to the MIB's interfaces objects.

10. Remove all SNMP v2c configurations.

SNMP v3

11. **SNMP version 3 allows the authentication and/or encryption of data.** Configure the SNMP v3 access by defining 4 different users and 4 different user groups to establish:

1) An access without authentication/encryption,
2) An access without authentication/encryption but with view limitations,
3) An access with authentication (MD5) and no encryption,
4) An access with authentication (MD5) and encryption (DES56).

```
Router(config)# snmp-server engineID local 123456789A
Router(config)# snmp-server user user1 group1 v3
Router(config)# snmp-server user user2 group2 v3
Router(config)# snmp-server user user3 group3 v3 auth md5 authpass
Router(config)# snmp-server user user4 group4 v3 auth md5 authpass priv des56 encpassword
Router(config)# snmp-server group group1 v3 noauth
Router(config)# snmp-server group group2 v3 noauth read myview
Router(config)# snmp-server group group3 v3 auth
Router(config)# snmp-server group group4 v3 priv
Router(config)# snmp-server view myview system included
Router(config)# snmp-server community myrouter RO
```

Use the following commands to verify the SNMP v3 users/groups information:

```
Router# show snmp user
Router# show snmp group
```

12. Start a capture with Wireshark, test the following SNMP v3 requests and analyze the outputs and captured SNMP packets:

```
snmpwalk -v1 -c myrouter 192.1.1.#group sysDescr
snmpwalk -v2c -c myrouter 192.1.1.#group sysDescr
snmpwalk -v3 -u user1 -l noauthnopriv 192.1.1.#group
snmpwalk -v3 -u user2 -l noauthnopriv 192.1.1.#group
snmpwalk -v3 -u user1 -l noauthnopriv 192.1.1.#group sysDescr
snmpwalk -v3 -u user2 -l noauthnopriv 192.1.1.#group sysDescr
snmpwalk -v3 -u user1 -l noauthnopriv 192.1.1.#group interfaces
snmpwalk -v3 -u user2 -l noauthnopriv 192.1.1.#group interfaces
snmpwalk -v3 -u user3 -l authnopriv 192.1.1.#group sysDescr
snmpwalk -v3 -u user3 -A authpass -l authnopriv 192.1.1.#group sysDescr
snmpwalk -v3 -u user4 -A authpass -l authpriv 192.1.1.#group sysDescr
snmpwalk -v3 -u user4 -A authpass -X encpassword -l authpriv 192.1.1.#group sysDescr
```

SNMP traps

13. Routers can generate automatic SNMP messages to notify a specif event (SNMP traps). Perform the following configurations to generate a SNMP trap every time the Router's system log has a new entry:

```
Router(config)# snmp-server enable traps syslog
Router(config)# snmp-server host 192.1.1.100+#group version 2c myrouter
```

Start a capture with Wireshark. At the router, change (several times) the status of the Ethernet interface connected to 192.#group.10.0/24. Analyze the captured packets.

Extra – Monitoring/Management Scripts

14. Connect your hub to the rack in order to interconnect your PC and Router to all other Routers in the room. Define the following bash script (getRouter.sh) to retrieve all Routers model and firmware version.

```
#!/bin/bash
for i in {1..9}
do
     model=`snmpget -v2c -c myrouter 192.1.1.$i SNMPv2-SMI::mib-2.47.1.1.1.1.2.1 | cut -d"\"" -f2`
     if [ -n "$model" ]
     then
      fwver=`snmpget -v2c -c myrouter 192.1.1.$i SNMPv2-MIB::sysDescr.0 | head -n1 | cut -d"," -f2,3`
      echo "Router 192.1.1.$i, $model, $fwver"
     fi
done
```

Run using the command `bash getRouter.sh`.

15. Define a bash script (getTraff.sh) to retrieve all Routers interfaces and respective transmitted traffic (in bytes). Run using the command `bash getTraff.sh`.

16. Define a bash script to detect IP spoofing attacks in LANs (getSpoff.sh) that periodically retrieve all Routers ARP tables and detect MAC changes for the same IP address.