COSC 3364 – Principles of Cybersecurity
Lab 09

Provide screenshots where * is indicated.

Perform:

```
sudo apt install net-tools

sudo apt install nmap

sudo apt install snort
```

Network Configuration

One of the commonly used commands to display network information is the **ifconfig** command. When executed with no arguments, it lists active network devices.

Many different flags can be assigned an interface. Some of the more important flags include the following:

> UP: Indicates the interface is active. When the interface is down, the flags line is not displayed at all.
>
> BROADCAST: Indicates that the broadcast address has been set for the device.
>
> MULTICAST: Indicates whether the multicast address is enabled on this device.
>
> PROMISC: Indicates whether the device is in promiscuous mode. Normally a device only listens to network packets sent to its own IP address. In promiscuous mode, the device listens for all network traffic. This can be helpful for analyzing network traffic.

Enabling promiscuous mode allows you to *sniff* the network. This means you can observe network traffic either to determine issues or to discover a potential security breech.

The **arp** command is used to view the ARP table or make changes to it. When executed with no arguments, the **arp** command displays the ARP table

The **route** command either displays or modifies the routing table. To display the routing table, execute the **route** command without any arguments.

The **netstat** command is useful for displaying a variety of network information. It is a key utility when troubleshooting network issues.

Option	Description
-t or –tcp	Display TCP information.
-u or –udp	Display UDP information.
-r or –route	Display the routing table.
-v or –verbose	Verbose; display additional information.
-i or –interfaces	Display information based on a specific interface.
-a or –all	Apply to all.
-s or –statistics	Display statistics for the output.

1. Display network information*

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
labuser1@ML-RefVm-535928:~/Desktop$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.48  netmask 255.255.240.0  broadcast 10.0.15.255
        inet6 fe80::6245:bdff:fef7:6c65  prefixlen 64  scopeid 0x20<link>
        ether 60:45:bd:f7:6c:65  txqueuelen 1000  (Ethernet)
        RX packets 20075  bytes 12416253 (12.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16767  bytes 23790087 (23.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 97  bytes 13224 (13.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 97  bytes 13224 (13.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

labuser1@ML-RefVm-535928:~/Desktop$
```

2. Enable promiscuous mode on eth0

3. Display flags section for eht0*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo ifconfig eth0 promisc
labuser1@ML-RefVm-535928:~/Desktop$ ifconfig
eth0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
        inet 10.0.0.48  netmask 255.255.240.0  broadcast 10.0.15.255
        inet6 fe80::6245:bdff:fef7:6c65  prefixlen 64  scopeid 0x20<link>
        ether 60:45:bd:f7:6c:65  txqueuelen 1000  (Ethernet)
        RX packets 22109  bytes 12692781 (12.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 19160  bytes 25736587 (25.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4. Disable promiscuous mode on eth0

5. Display the IP routing table*

```
labuser1@ML-RefVm-535928:~/Desktop$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 eth0
10.0.0.0        0.0.0.0         255.255.240.0   U     100    0        0 eth0
_gateway        0.0.0.0         255.255.255.255 UH    100    0        0 eth0
168.63.129.16   _gateway        255.255.255.255 UGH   100    0        0 eth0
```

6. Display the ARP table*

```
labuser1@ML-RefVm-535928:~/Desktop$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.45                ether   12:34:56:78:9a:bc   C                     eth0
_gateway                 ether   12:34:56:78:9a:bc   C                     eth0
10.0.0.40                ether   12:34:56:78:9a:bc   C                     eth0
labuser1@ML-RefVm-535928:~/Desktop$
```

7. Display all tcp connections*

```
labuser1@ML-RefVm-535928:~/Desktop$ netstat -tcp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp6       0      0 ML-RefVm-:ms-wbt-server 173.255.36.1:57751      ESTABLISHED
-
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp6       0      0 ML-RefVm-:ms-wbt-server 173.255.36.1:57751      ESTABLISHED
-
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
^Z
[2]+  Stopped                 netstat -tcp
labuser1@ML-RefVm-535928:~/Desktop$
```

Footprinting

Footprinting, or reconnaissance, is the process of discovering information about a network or system with the intent to use this information to compromise security measures. A large variety of footprinting techniques and tools can provide useful information. This information is then used in conjunction with other hacking tactics to gain unauthorized access to a network or system.

Always make sure you have written consent to perform footprinting actions on any system in an organization. Just because you work for a company does not mean you are authorized to perform these actions. In most countries, the act of performing footprinting actions is illegal, and many organizations have prosecuted their own employees who were not authorized to perform these actions.

The **nmap** command is used to probe a remote system to determine which network ports are reachable from the local system. This is useful for many reasons:

- Determining what services are available on the remote system.
- Testing security features on the remote system, such as TCP wrappers.
- If the **nmap** command is executed from a remote network, the output could verify the effectiveness of your network's firewall.

To use the **nmap** command, provide either the IP address or hostname of the system you want to scan.

- Provide either the IP address or hostname of the system you want to scan
  - Example: nmap 192.168.1.1

- The lines that describe the open ports start with the port number/protocol and end with the corresponding service
    - Example: 23/tcp open telnet
- By default, only TCP ports are scanned
    - Use the -sU option to scan UDP ports
- By default, only certain common ports are scanned
    - Use -p followed by a range of port numbers to expand that
    - Example: nmap -p 1-65535 192.168.1.1
- Use the -sV option to see service version information
- Use the -sP option to find out what IP addresses are in use
- Use the --iflist option to see information about your own system, including a list of network interfaces and the routing table

1. Determine the network interfaces and routing table*

```
labuser1@ML-RefVm-535928:~/Desktop$ nmap --iflist
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 22:45 UTC
************************INTERFACES************************
DEV  (SHORT) IP/MASK                     TYPE     UP MTU   MAC
lo   (lo)    127.0.0.1/8                 loopback up 65536
lo   (lo)    ::1/128                     loopback up 65536
eth0 (eth0)  10.0.0.48/20                ethernet up 1500  60:45:BD:F7:6C:65
eth0 (eth0)  fe80::6245:bdff:fef7:6c65/64 ethernet up 1500  60:45:BD:F7:6C:65

**************************ROUTES**************************
DST/MASK                    DEV  METRIC GATEWAY
10.0.0.1/32                 eth0 100
168.63.129.16/32            eth0 100    10.0.0.1
169.254.169.254/32          eth0 100    10.0.0.1
10.0.0.0/20                 eth0 100
0.0.0.0/0                   eth0 100    10.0.0.1
::1/128                     lo   0
fe80::6245:bdff:fef7:6c65/128 eth0 0
::1/128                     lo   256
fe80::/64                   eth0 256
ff00::/8                    eth0 256

labuser1@ML-RefVm-535928:~/Desktop$
```

2. Scan the listed eth0 IPv4 routes to determine any open TCP ports.* (Only screenshot any with open ports)

```
labuser1@ML-RefVm-535928:~/Desktop$ nmap 0.0.0.0
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 22:47 UTC
Nmap scan report for 0.0.0.0
Host is up (0.00014s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3306/tcp open  mysql
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
labuser1@ML-RefVm-535928:~/Desktop$
```

3. Scan the listed eth0 IPv4 routes to determine any open UDP ports.* (Only screenshot any with open ports)

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo nmap -sU 0.0.0.0
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 22:48 UTC
Nmap scan report for 0.0.0.0
Host is up (0.0000050s latency).
Not shown: 999 filtered ports
PORT     STATE         SERVICE
5353/udp open|filtered zeroconf

Nmap done: 1 IP address (1 host up) scanned in 1.32 seconds
labuser1@ML-RefVm-535928:~/Desktop$
```

Intrusion Detection

Several intrusion detection tools are installed by default on most Linux distributions. A hacker who has unauthorized access to your system very likely has an established network connection. Probing your system on a regular basis can help you determine if an unauthorized user is accessing your system. Look for any unusual connections and pay attention to where these connections originate (the "Foreign Address" column).

Another **netstat** command you should consider running on a regular basis is the **netstat -taupe** command. This command displays all open ports, which is important because hackers often will open new ports to create more backdoors into the system. You should be aware of what ports should be open on each system in your network, and routinely verify that the correct ports are open and that no additional ports are open on each system.

Another useful intrusion detection tool is the **tcpdump** command. This tool allows you to probe network traffic, searching for any suspicious activity. For your purposes, you should use the command within your intrusion detection game plan to warn you of any rogue access points or other unauthorized hardware. By

default, the **tcpdump** command displays all network traffic to standard output until you terminate the command. This could result in a dizzying amount of data flying by on your screen.

You can limit the output to a specific number of network packets by using the **-c** option. More likely you want to capture the output based on some sort of criteria. For example, you can have the **tcpdump** command only capture packets available on a specific interface by using the **-i** option and to limit packets to only a specific protocol, indicate the protocol name as an argument. To only display packets associated with a specific port, use the **port** argument. You can also limit the packets based on the source IP or destination IP.

1. Display all open ports.*

```
labuser1@ML-RefVm-535928:~/Desktop$ netstat -taupe
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User       Inode      PID/Program name
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
systemd-resolve 4039       -
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
root       5544       -
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
mysql      6158       -
tcp6       0      0 [::]:ms-wbt-server      [::]:*                  LISTEN
xrdp       5004       -
tcp6       0      0 ip6-localhost:3350      [::]:*                  LISTEN
root       4530       -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
root       4643       -
tcp6       0      0 ML-RefVm-:ms-wbt-server 173.255.36.1:57751      ESTABLISHED
xrdp       10730      -
udp        0      0 localhost:323           0.0.0.0:*
root       4561       -
udp        0      0 ML-RefVm-535928:37699   168.63.129.16:domain    ESTABLISHED
systemd-resolve 22450      -
udp        0      0 0.0.0.0:35960           0.0.0.0:*
avahi      4524       -
udp      896      0 localhost:50299         localhost:domain        ESTABLISHED
labuser1   24427      2622/netstat
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
avahi      4522       -
udp      896      0 localhost:52675         localhost:domain        ESTABLISHED
labuser1   24322      2621/netstat
udp        0      0 localhost:domain        0.0.0.0:*
systemd-resolve 4038       -
udp        0      0 ML-RefVm-535928:bootpc  0.0.0.0:*
systemd-network 2816       -
udp6       0      0 ip6-localhost:323       [::]:*
root       4562       -
udp6       0      0 [::]:59915              [::]:*
avahi      4525       -
udp6       0      0 [::]:mdns               [::]:*
avahi      4523       -
labuser1@ML-RefVm-535928:~/Desktop$
```

2. Display network traffic.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:51:07.491772 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 939774207:939774356, ack 2112141472, win 1026, length 149
22:51:07.513286 IP 173.255.36.1.57751 > ML-RefVm-535928.ms-wbt-server: Flags [P.
], seq 1:27, ack 0, win 1026, length 26
22:51:07.513334 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [.]
, ack 27, win 1026, length 0
22:51:07.532554 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 149:3069, ack 27, win 1026, length 2920
22:51:07.532578 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 3069:5989, ack 27, win 1026, length 2920
22:51:07.532968 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 5989:8909, ack 27, win 1026, length 2920
22:51:07.532980 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 8909:11829, ack 27, win 1026, length 2920
22:51:07.560733 IP 173.255.36.1.57751 > ML-RefVm-535928.ms-wbt-server: Flags [.]
, ack 5989, win 1029, length 0
22:51:07.560786 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
```

```
22:51:19.804341 IP 168.63.129.16.http > ML-RefVm-535928.52280: Flags [.], ack 38
1, win 16387, options [nop,nop,TS val 3752555700 ecr 2138889678], length 0
^C22:51:19.804493 IP ML-RefVm-535928.49560 > 169.254.169.254.http: Flags [S], se
q 4246620783, win 64240, options [mss 1460,sackOK,TS val 2154303385 ecr 0,nop,ws
cale 7], length 0

1676 packets captured
2220 packets received by filter
160 packets dropped by kernel
labuser1@ML-RefVm-535928:~/Desktop$
```

3. Display 5 packets of network traffic on eth0.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo tcpdump -i eth0 -c 5
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:53:02.449821 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 959339983:959340143, ack 2112159595, win 1186, length 160
22:53:02.485150 IP 173.255.36.1.57751 > ML-RefVm-535928.ms-wbt-server: Flags [P.
], seq 1:27, ack 160, win 1026, length 26
22:53:02.485199 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [.]
, ack 27, win 1186, length 0
22:53:02.507765 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 160:3080, ack 27, win 1186, length 2920
22:53:02.513130 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 3080:6000, ack 27, win 1186, length 2920
5 packets captured
584 packets received by filter
176 packets dropped by kernel
labuser1@ML-RefVm-535928:~/Desktop$
```

4. Display 5 packets of only TCP network traffic on eth0.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo tcpdump -i eth0 -c 5 tcp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:53:48.561354 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 960290837:960290988, ack 2112163635, win 1259, length 151
22:53:48.580655 IP 173.255.36.1.57751 > ML-RefVm-535928.ms-wbt-server: Flags [P.
], seq 1:27, ack 0, win 1028, length 26
22:53:48.580692 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [.]
, ack 27, win 1259, length 0
22:53:48.613227 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 151:3071, ack 27, win 1259, length 2920
22:53:48.613248 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.57751: Flags [P.
], seq 3071:5991, ack 27, win 1259, length 2920
5 packets captured
514 packets received by filter
111 packets dropped by kernel
labuser1@ML-RefVm-535928:~/Desktop$
```

5. Display 5 packets of network traffic on port 80 and eth0.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo tcpdump -i eth0 port 80 -c 5
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:55:20.019686 IP ML-RefVm-535928.34352 > 168.63.129.16.http: Flags [S], seq 89
283134, win 64240, options [mss 1460,sackOK,TS val 2139129893 ecr 0,nop,wscale 7
], length 0
22:55:20.020327 IP 168.63.129.16.http > ML-RefVm-535928.34352: Flags [S.], seq 5
79221189, ack 89283135, win 65535, options [mss 1460,nop,wscale 8,sackOK,TS val
3752795914 ecr 2139129893], length 0
22:55:20.020368 IP ML-RefVm-535928.34352 > 168.63.129.16.http: Flags [.], ack 1,
 win 502, options [nop,nop,TS val 2139129894 ecr 3752795914], length 0
22:55:20.020471 IP ML-RefVm-535928.34352 > 168.63.129.16.http: Flags [P.], seq 1
:201, ack 1, win 502, options [nop,nop,TS val 2139129894 ecr 3752795914], length
 200: HTTP: GET /machine/?comp=goalstate HTTP/1.1
22:55:20.022943 IP 168.63.129.16.http > ML-RefVm-535928.34352: Flags [FP.], seq
1:2312, ack 201, win 16387, options [nop,nop,TS val 3752795917 ecr 2139129894],
length 2311: HTTP: HTTP/1.1 200 OK
5 packets captured
34 packets received by filter
0 packets dropped by kernel
labuser1@ML-RefVm-535928:~/Desktop$
```

Snort

https://docs.snort.org/welcome

1. Capture on local interface with Snort comparing with **-c** to configuration file `/etc/snort/snort.conf*`

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo snort -i eth0 -c /etc/snort/snort.conf
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 901 1220 1
14 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7
45 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8
43 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:3444
 41080 50002 55555 ]
```

2. Navigate to `/etc/snort/rules/local.rules`

   Add rule:

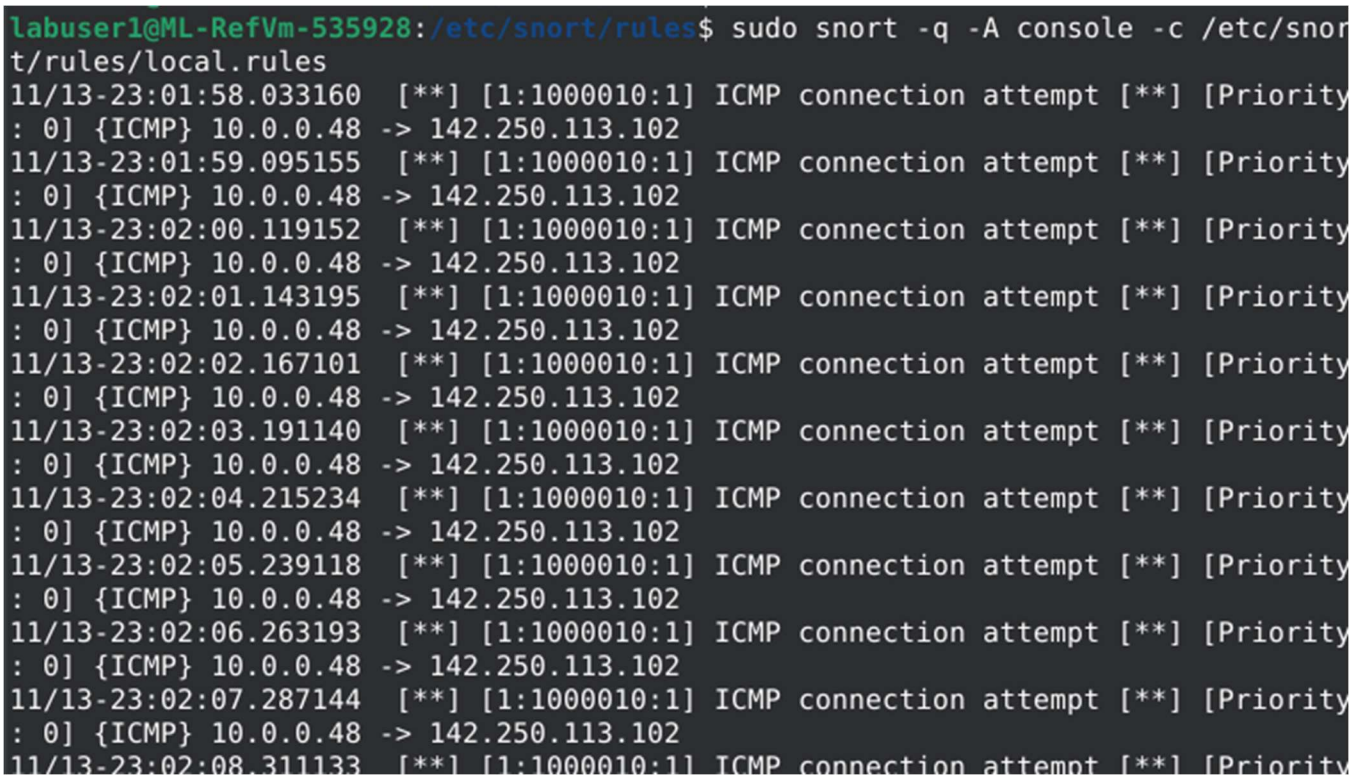
   ```
   alert icmp any any -> any any (msg:"ICMP connection attempt"; sid:1000010;
   rev:1;)
   ```

3. Test the newly updated local.rules with:

   ```
   snort -q -A console -c /etc/snort/rules/local.rules
   ```

4. In another terminal use the command **ping** to any hostname and monitor the alerts.*

```
labuser1@ML-RefVm-535928:/etc/snort/rules$ ping google.com
PING google.com (142.250.113.102) 56(84) bytes of data.
```


labuser1@ML-RefVm-535928:/etc/snort/rules$ sudo snort -q -A console -c /etc/snor
t/rules/local.rules
11/13-23:01:58.033160 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:01:59.095155 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:00.119152 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:01.143195 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:02.167101 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:03.191140 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:04.215234 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:05.239118 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:06.263193 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102
11/13-23:02:07.287144 [**] [1:1000010:1] ICMP connection attempt [**] [Priority
: 0] {ICMP} 10.0.0.48 -> 142.250.113.102