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:

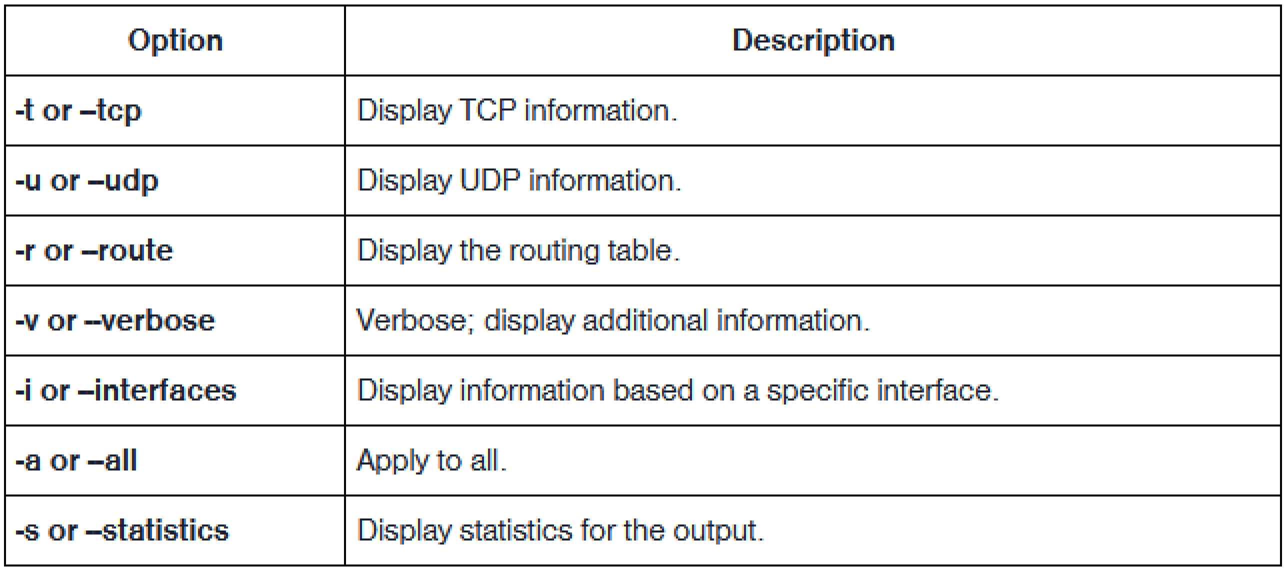
> UP: Indicates the interface is active. When the interface is down, the flags line is not displayed at all.
>
> BROADCAST: Indicates that the broadcast address has been set for the device.
>
> MULTICAST: Indicates whether the multicast address is enabled on this device.
>
> PROMISC: Indicates whether 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

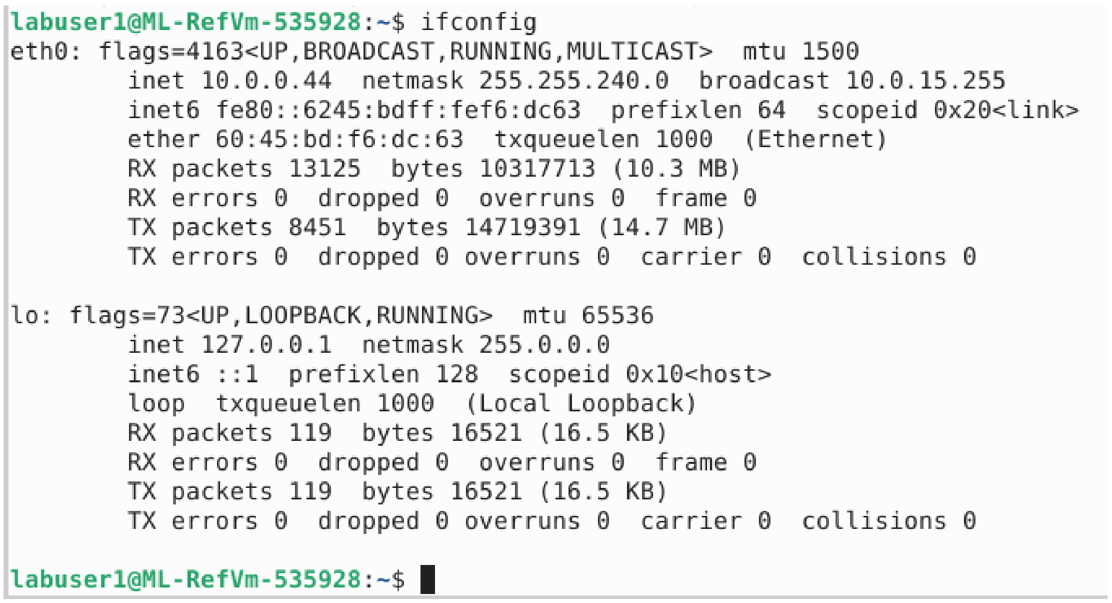
The **route** command either displays or modifies the routing table. To display the routing table, execute the **route** command without any arguments.

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*



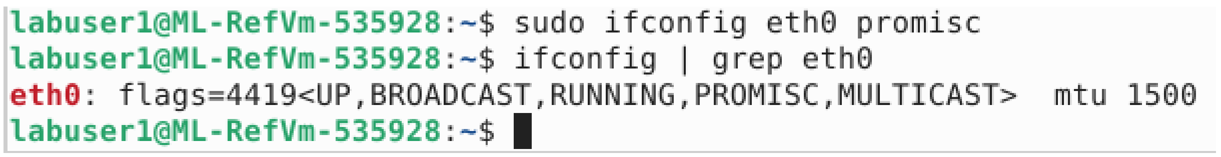
```
labuser1@ML-RefVm-535928:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.44  netmask 255.255.240.0  broadcast 10.0.15.255
        inet6 fe80::6245:bdff:fef6:dc63  prefixlen 64  scopeid 0x20<link>
        ether 60:45:bd:f6:dc:63  txqueuelen 1000  (Ethernet)
        RX packets 13125  bytes 10317713 (10.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8451  bytes 14719391 (14.7 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 119  bytes 16521 (16.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 119  bytes 16521 (16.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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

2. Enable promiscuous mode on eth0

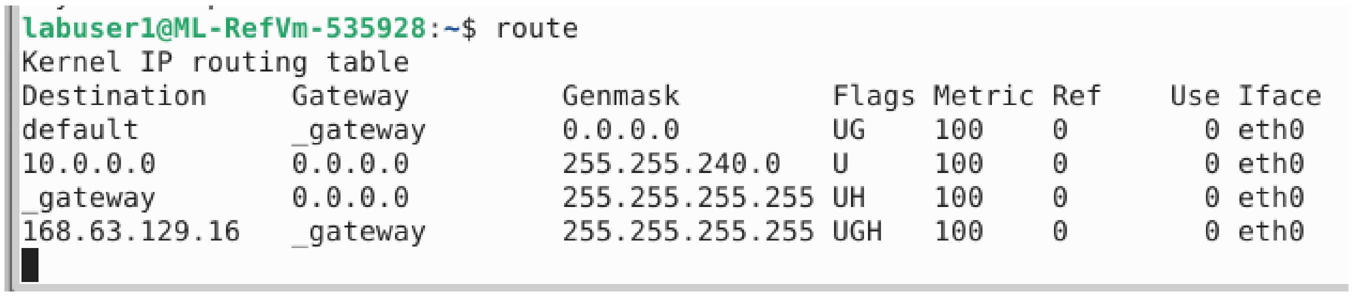
3. Display flags section for eht0*



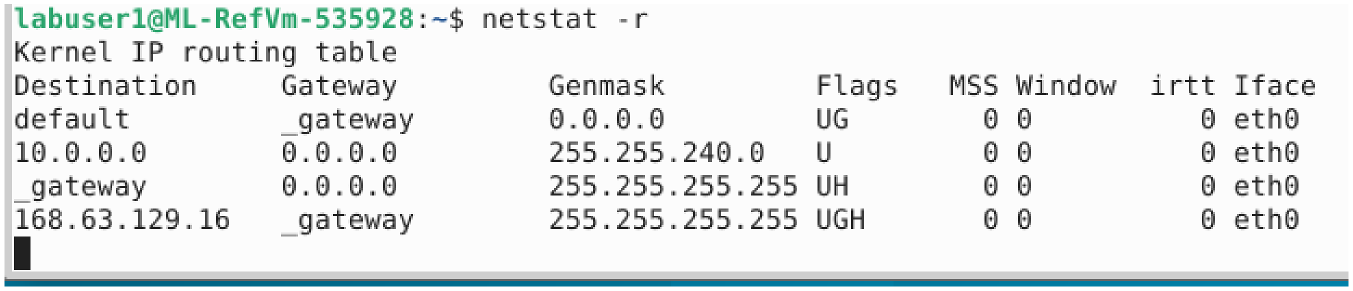
```
labuser1@ML-RefVm-535928:~$ sudo ifconfig eth0 promisc
labuser1@ML-RefVm-535928:~$ ifconfig | grep eth0
eth0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>  mtu 1500
labuser1@ML-RefVm-535928:~$ 
```

4. Disable promiscuous mode on eth0

5. Display the IP routing table*

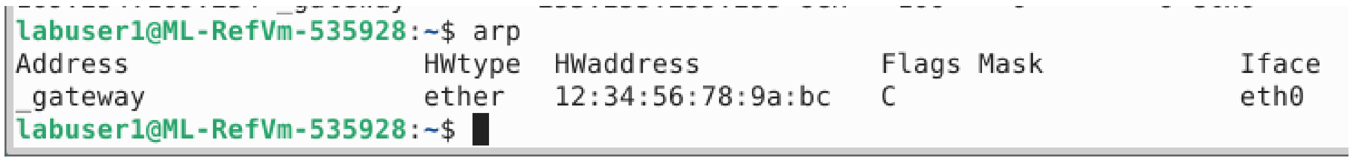


```
labuser1@ML-RefVm-535928:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 eth0
10.0.0.0        0.0.0.0         255.255.240.0   U     100    0        0 eth0
_gateway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



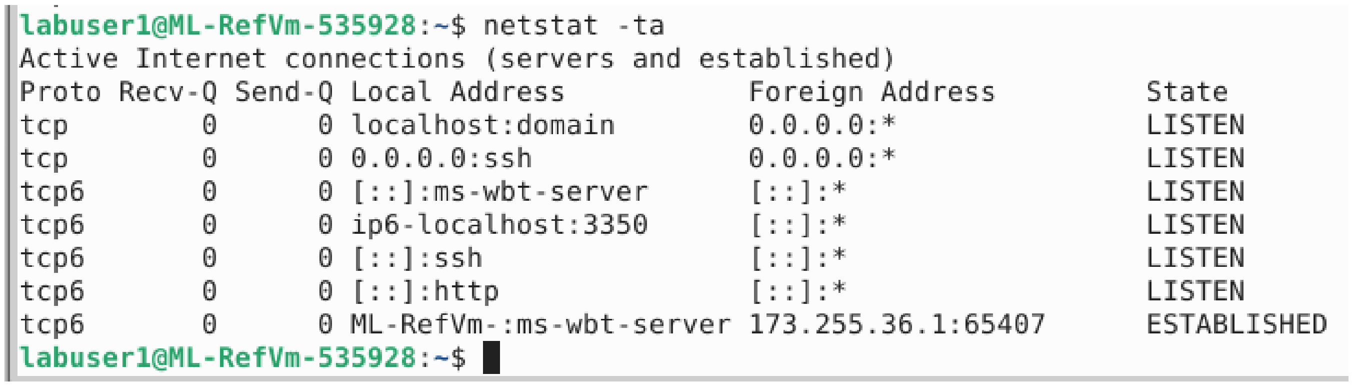
```
labuser1@ML-RefVm-535928:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 eth0
10.0.0.0        0.0.0.0         255.255.240.0   U         0 0          0 eth0
_gateway        0.0.0.0         255.255.255.255 UH        0 0          0 eth0
168.63.129.16   _gateway        255.255.255.255 UGH       0 0          0 eth0
```

6. Display the ARP table*



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

7. Display all tcp connections*



```
labuser1@ML-RefVm-535928:~$ netstat -ta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ms-wbt-server      [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:3350      [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:http               [::]:*                  LISTEN
tcp6       0      0 ML-RefVm-:ms-wbt-server 173.255.36.1:65407      ESTABLISHED
labuser1@ML-RefVm-535928:~$
```

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.

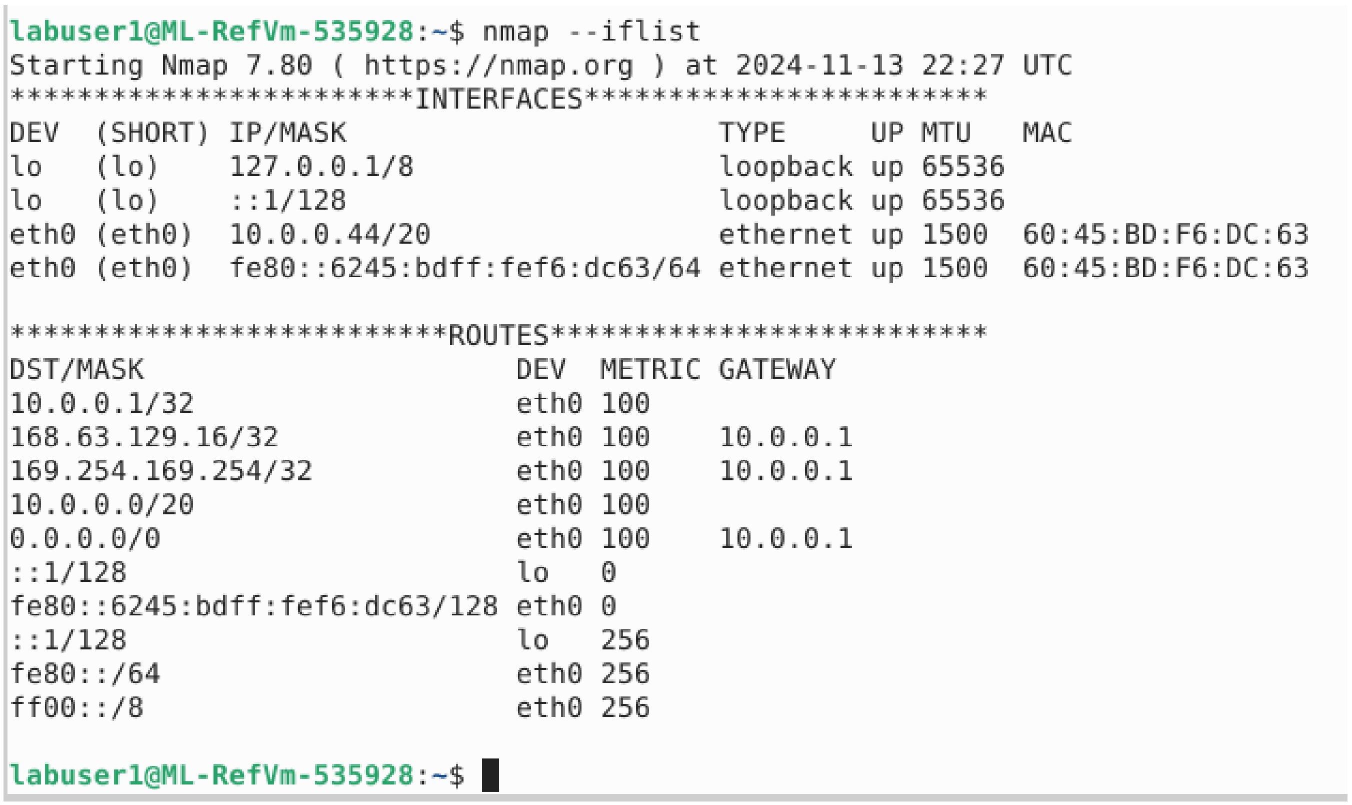
The **nmap** command is used to probe a remote system to determine which network ports are reachable from the local system. This is useful for many reasons:

- 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*

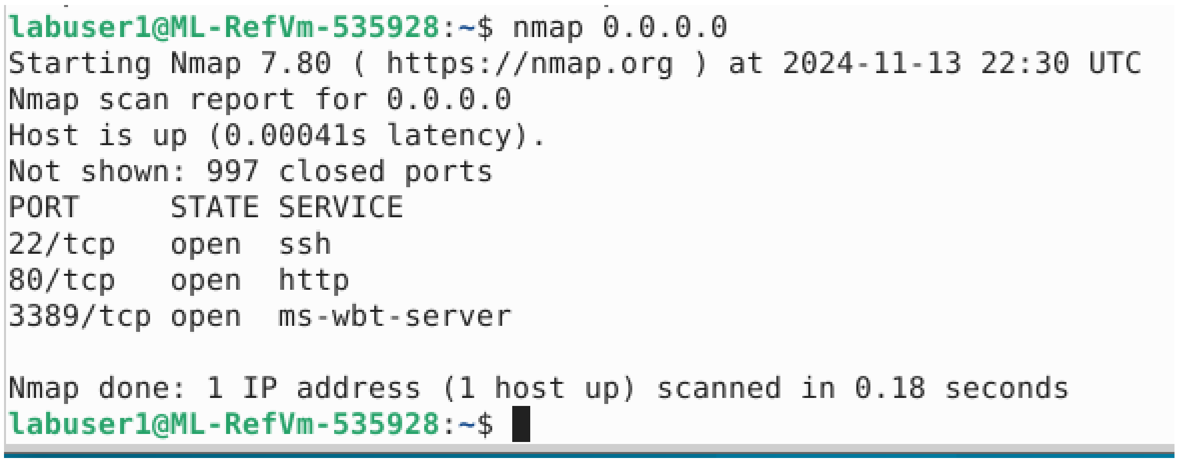


```
labuser1@ML-RefVm-535928:~$ nmap --iflist
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 22:27 UTC
************************INTERFACES************************
DEV  (SHORT) IP/MASK                      TYPE     UP MTU   MAC
lo   (lo)    127.0.0.1/8                  loopback up 65536
lo   (lo)    ::1/128                      loopback up 65536
eth0 (eth0)  10.0.0.44/20                 ethernet up 1500  60:45:BD:F6:DC:63
eth0 (eth0)  fe80::6245:bdff:fef6:dc63/64 ethernet up 1500  60:45:BD:F6:DC:63

**************************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:fef6:dc63/128 eth0 0
::1/128                       lo   256
fe80::/64                     eth0 256
ff00::/8                      eth0 256

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

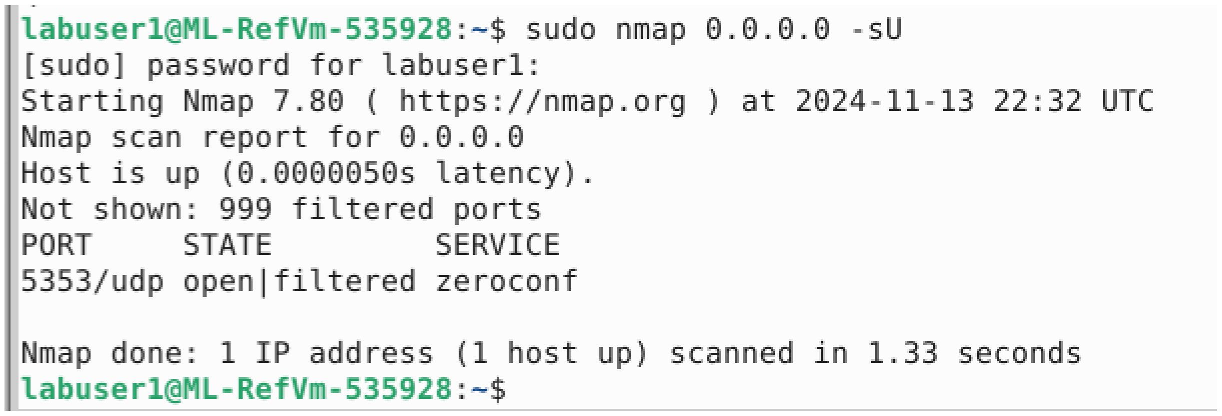
2. Scan the listed eth0 IPv4 routes to determine any open TCP ports.* (Only screenshot any with open ports)



```
labuser1@ML-RefVm-535928:~$ nmap 0.0.0.0
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 22:30 UTC
Nmap scan report for 0.0.0.0
Host is up (0.00041s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
labuser1@ML-RefVm-535928:~$
```

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



```
labuser1@ML-RefVm-535928:~$ sudo nmap 0.0.0.0 -sU
[sudo] password for labuser1:
Starting Nmap 7.80 ( https://nmap.org ) at 2024-11-13 22:32 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 1.33 seconds
labuser1@ML-RefVm-535928:~$
```

Intrusion Detection

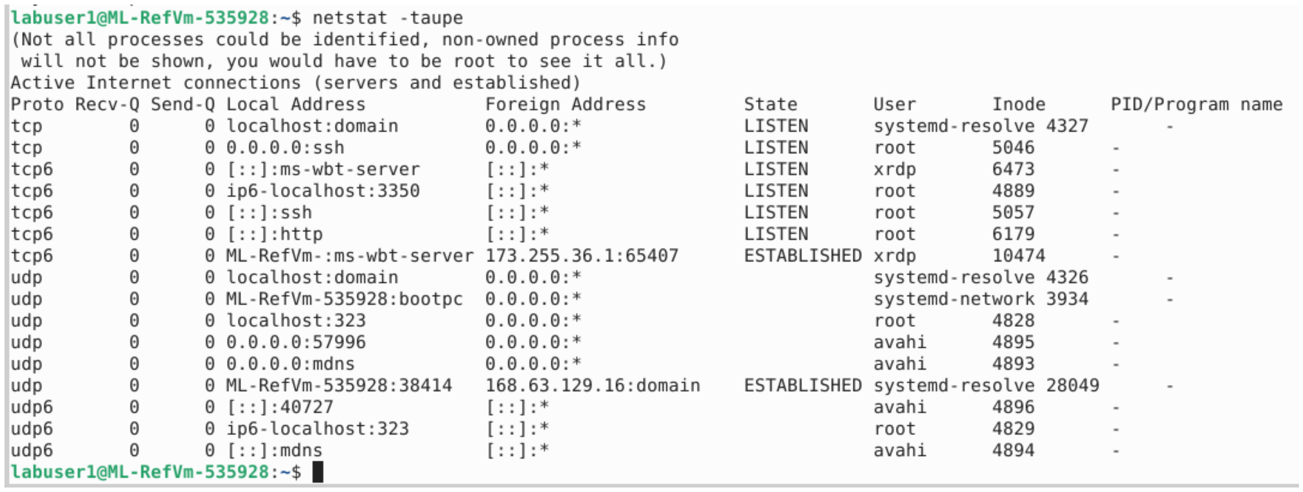
Several intrusion detection tools are installed by default on most Linux distributions. A hacker who has unauthorized access to your system very likely has an established network connection. Probing your system on a regular basis can help you determine if an unauthorized user is accessing your system. 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.

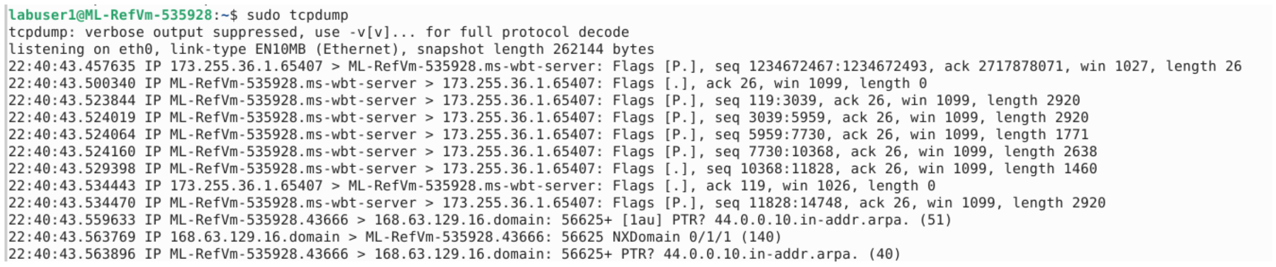
You can limit the output to a specific number of network packets by using the **-c** option. More likely you want to capture the output based on some sort of criteria. For example, you can have the **tcpdump** command only capture packets available on a specific interface by using the **-i** option and 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.*



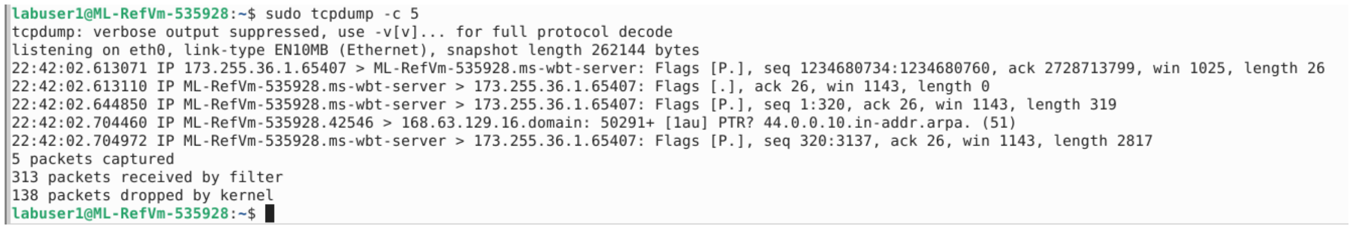
```
labuser1@ML-RefVm-535928:~$ netstat -taupe
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User       Inode      PID/Program name
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      systemd-resolve 4327       -
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      root       5046       -
tcp6       0      0 [::]:ms-wbt-server      [::]:*                  LISTEN      xrdp       6473       -
tcp6       0      0 ip6-localhost:3350      [::]:*                  LISTEN      root       4889       -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      root       5057       -
tcp6       0      0 [::]:http               [::]:*                  LISTEN      root       6179       -
tcp6       0      0 ML-RefVm-:ms-wbt-server 173.255.36.1:65407      ESTABLISHED xrdp       10474      -
udp        0      0 localhost:domain        0.0.0.0:*                           systemd-resolve 4326       -
udp        0      0 ML-RefVm-535928:bootpc  0.0.0.0:*                           systemd-network 3934       -
udp        0      0 localhost:323           0.0.0.0:*                           root       4828       -
udp        0      0 0.0.0.0:57996           0.0.0.0:*                           avahi      4895       -
udp        0      0 0.0.0.0:mdns            0.0.0.0:*                           avahi      4893       -
udp        0      0 ML-RefVm-535928:38414   168.63.129.16:domain    ESTABLISHED systemd-resolve 28049      -
udp6       0      0 [::]:40727              [::]:*                              avahi      4896       -
udp6       0      0 ip6-localhost:323       [::]:*                              root       4829       -
udp6       0      0 [::]:mdns               [::]:*                              avahi      4894       -
labuser1@ML-RefVm-535928:~$ █
```

2. Display network traffic.*



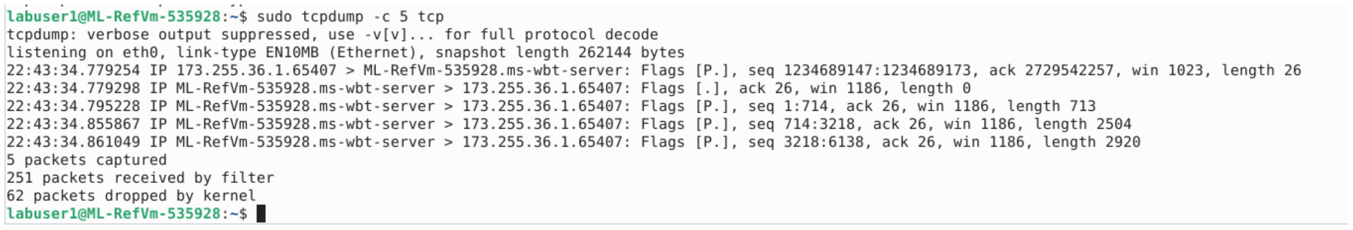
```
labuser1@ML-RefVm-535928:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:40:43.457635 IP 173.255.36.1.65407 > ML-RefVm-535928.ms-wbt-server: Flags [P.], seq 1234672467:1234672493, ack 2717878071, win 1027, length 26
22:40:43.500340 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [.], ack 26, win 1099, length 0
22:40:43.523844 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 119:3039, ack 26, win 1099, length 2920
22:40:43.524019 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 3039:5959, ack 26, win 1099, length 2920
22:40:43.524064 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 5959:7730, ack 26, win 1099, length 1771
22:40:43.524160 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 7730:10368, ack 26, win 1099, length 2638
22:40:43.529398 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [.], seq 10368:11828, ack 26, win 1099, length 1460
22:40:43.534443 IP 173.255.36.1.65407 > ML-RefVm-535928.ms-wbt-server: Flags [.], ack 119, win 1026, length 0
22:40:43.534470 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 11828:14748, ack 26, win 1099, length 2920
22:40:43.559633 IP ML-RefVm-535928.43666 > 168.63.129.16.domain: 56625+ [1au] PTR? 44.0.0.10.in-addr.arpa. (51)
22:40:43.563769 IP 168.63.129.16.domain > ML-RefVm-535928.43666: 56625 NXDomain 0/1/1 (140)
22:40:43.563896 IP ML-RefVm-535928.43666 > 168.63.129.16.domain: 56625+ PTR? 44.0.0.10.in-addr.arpa. (40)
```

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



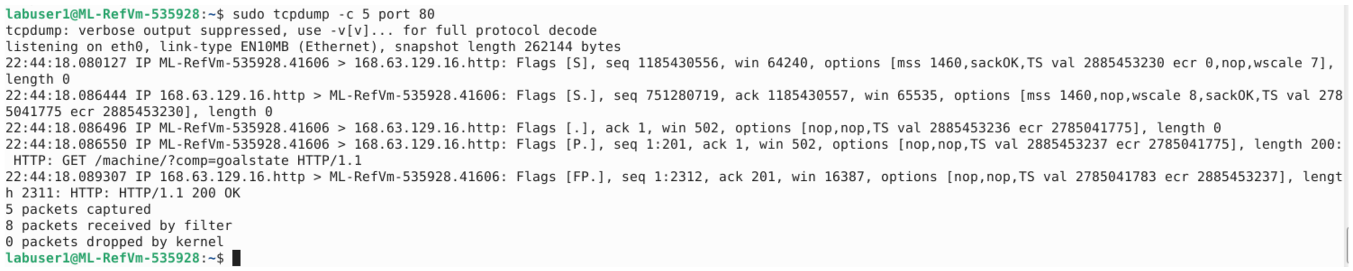
```
labuser1@ML-RefVm-535928:~$ sudo tcpdump -c 5
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:42:02.613071 IP 173.255.36.1.65407 > ML-RefVm-535928.ms-wbt-server: Flags [P.], seq 1234680734:1234680760, ack 2728713799, win 1025, length 26
22:42:02.613110 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [.], ack 26, win 1143, length 0
22:42:02.644850 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 1:320, ack 26, win 1143, length 319
22:42:02.704460 IP ML-RefVm-535928.42546 > 168.63.129.16.domain: 50291+ [1au] PTR? 44.0.0.10.in-addr.arpa. (51)
22:42:02.704972 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 320:3137, ack 26, win 1143, length 2817
5 packets captured
313 packets received by filter
138 packets dropped by kernel
labuser1@ML-RefVm-535928:~$ █
```

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



```
labuser1@ML-RefVm-535928:~$ sudo tcpdump -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
22:43:34.779254 IP 173.255.36.1.65407 > ML-RefVm-535928.ms-wbt-server: Flags [P.], seq 1234689147:1234689173, ack 2729542257, win 1023, length 26
22:43:34.779298 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [.], ack 26, win 1186, length 0
22:43:34.795228 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 1:714, ack 26, win 1186, length 713
22:43:34.855867 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 714:3218, ack 26, win 1186, length 2504
22:43:34.861049 IP ML-RefVm-535928.ms-wbt-server > 173.255.36.1.65407: Flags [P.], seq 3218:6138, ack 26, win 1186, length 2920
5 packets captured
251 packets received by filter
62 packets dropped by kernel
labuser1@ML-RefVm-535928:~$ █
```

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

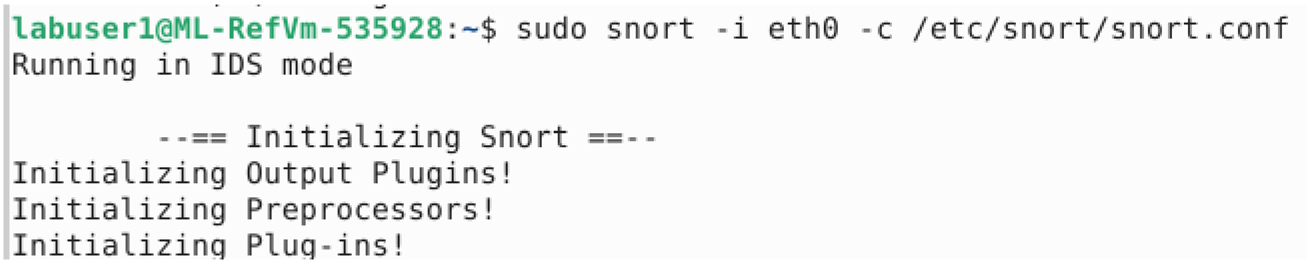


```
labuser1@ML-RefVm-535928:~$ sudo tcpdump -c 5 port 80
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:44:18.080127 IP ML-RefVm-535928.41606 > 168.63.129.16.http: Flags [S], seq 1185430556, win 64240, options [mss 1460,sackOK,TS val 2885453230 ecr 0,nop,wscale 7],
length 0
22:44:18.086444 IP 168.63.129.16.http > ML-RefVm-535928.41606: Flags [S.], seq 751280719, ack 1185430557, win 65535, options [mss 1460,nop,wscale 8,sackOK,TS val 278
5041775 ecr 2885453230], length 0
22:44:18.086496 IP ML-RefVm-535928.41606 > 168.63.129.16.http: Flags [.], ack 1, win 502, options [nop,nop,TS val 2885453236 ecr 2785041775], length 0
22:44:18.086550 IP ML-RefVm-535928.41606 > 168.63.129.16.http: Flags [P.], seq 1:201, ack 1, win 502, options [nop,nop,TS val 2885453237 ecr 2785041775], length 200:
 HTTP: GET /machine/?comp=goalstate HTTP/1.1
22:44:18.089307 IP 168.63.129.16.http > ML-RefVm-535928.41606: Flags [FP.], seq 1:2312, ack 201, win 16387, options [nop,nop,TS val 2785041783 ecr 2885453237], lengt
h 2311: HTTP: HTTP/1.1 200 OK
5 packets captured
8 packets received by filter
0 packets dropped by kernel
labuser1@ML-RefVm-535928:~$ █
```

Snort

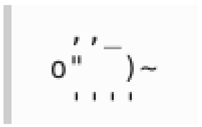
https://docs.snort.org/welcome

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



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

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
```



```
   ,,_
  o"  )~
   ''''
```

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

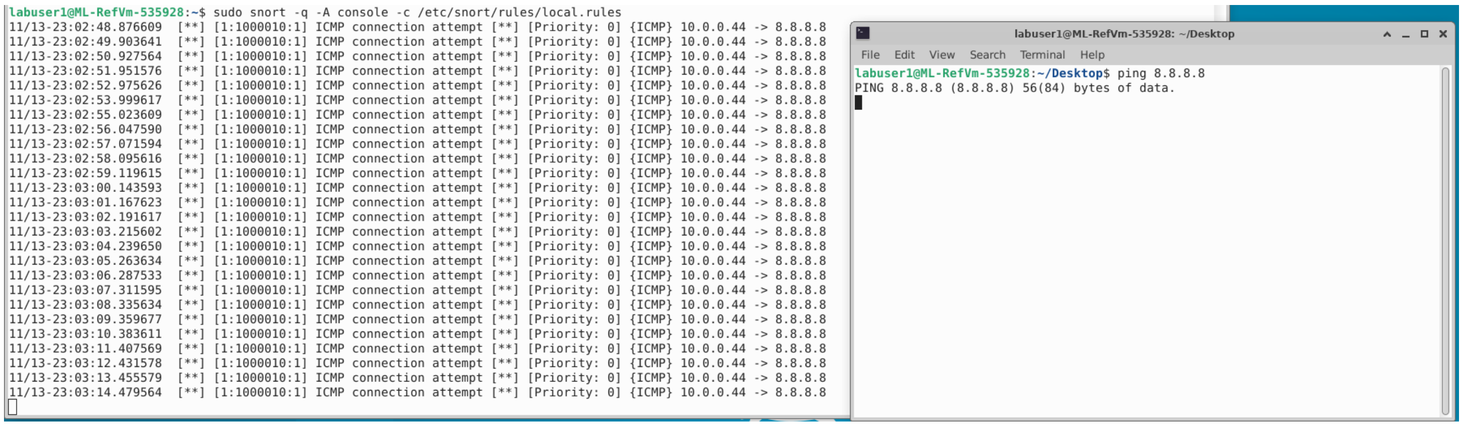
   Add rule:

   ```
   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:~$ sudo snort -q -A console -c /etc/snort/rules/local.rules
11/13-23:02:48.876609  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:49.903641  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:50.927564  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:51.951576  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:52.975626  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:53.999617  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:55.023609  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:56.047590  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:57.071594  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:58.095616  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:02:59.119615  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:00.143593  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:01.167623  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:02.191617  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:03.215602  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:04.239650  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:05.263634  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:06.287533  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:07.311595  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:08.335634  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:09.359677  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:10.383611  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:11.407569  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:12.431578  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:13.455579  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
11/13-23:03:14.479564  [**] [1:1000010:1] ICMP connection attempt [**] [Priority: 0] {ICMP} 10.0.0.44 -> 8.8.8.8
```

labuser1@ML-RefVm-535928: ~/Desktop
File Edit View Search Terminal Help

```
labuser1@ML-RefVm-535928:~/Desktop$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
```