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

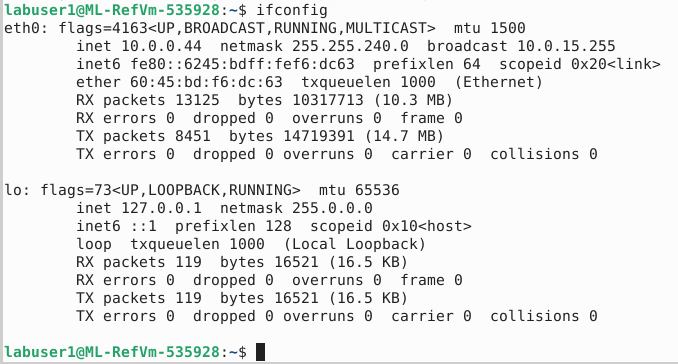
The **route** command either displays or modifies the routing table. To display the routing table, execute the **route** command without any arguments.

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

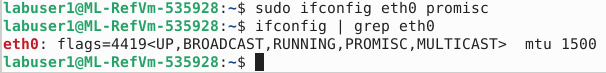
A white and black text

Description automatically generated with medium confidence

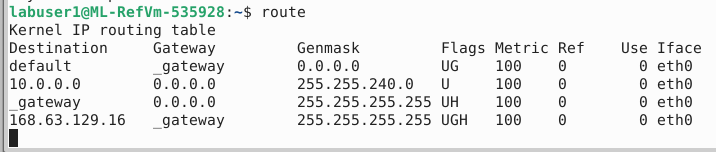
1. Display network information\*

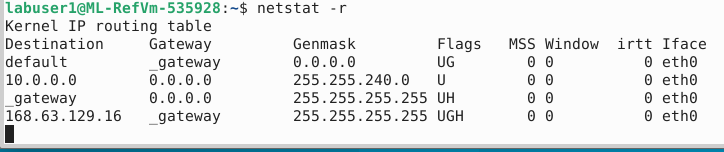


1. Enable promiscuous mode on eth0
2. Display flags section for eht0\*

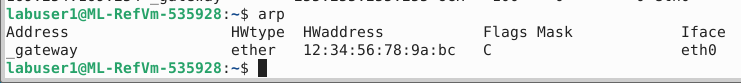


1. Disable promiscuous mode on eth0
2. Display the IP routing table\*

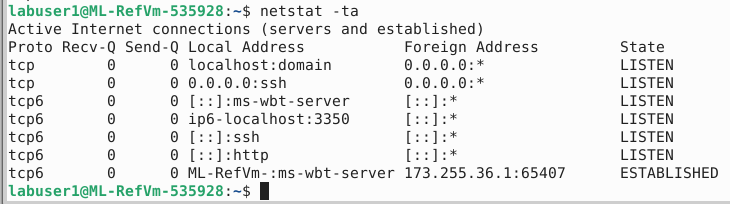




1. Display the ARP table\*



1. Display all tcp connections\*



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.

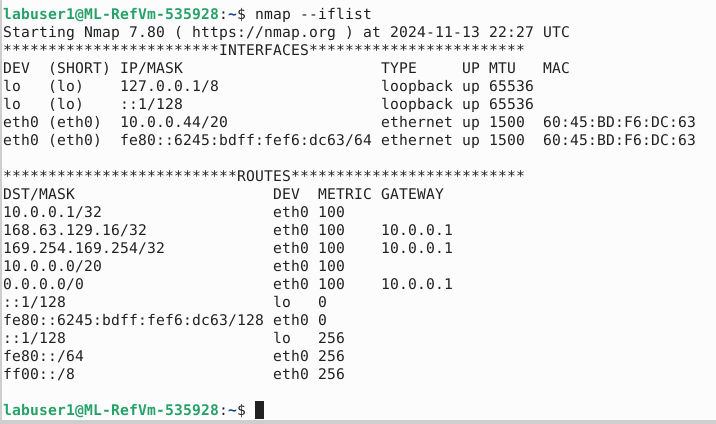
The **nmap** command is used to probe a remote system to determine which network ports are reachable from the local system. This is useful for many reasons:

* 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.

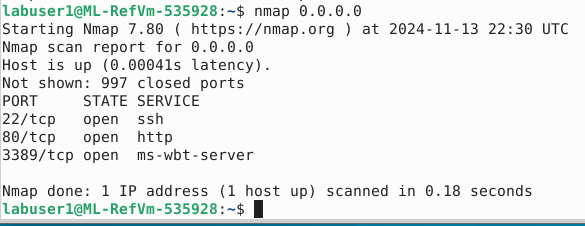
To use the **nmap** command, provide either the IP address or hostname of the system you want to scan.

* 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

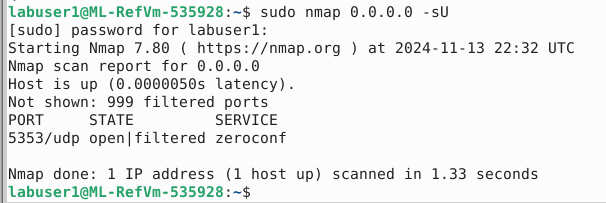
1. Determine the network interfaces and routing table\*



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



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



Intrusion Detection

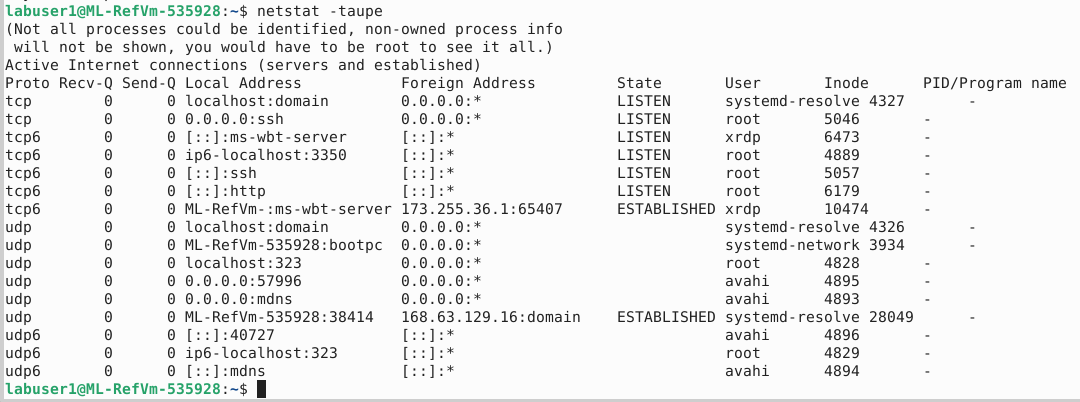
Several intrusion detection tools are installed by default on most Linux distributions. A hacker who has unauthorized access to your system very likely has an established network connection. Probing your system on a regular basis can help you determine if an unauthorized user is accessing your system.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

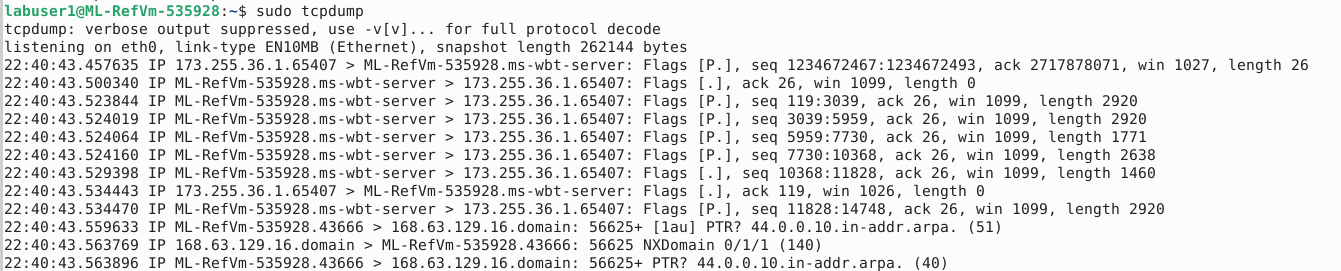
Another useful intrusion detection tool is the **tcpdump** command. This tool allows you to probe network traffic, searching for any suspicious activity. For your purposes, you should use the command within your intrusion detection game plan to warn you of any rogue access points or other unauthorized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

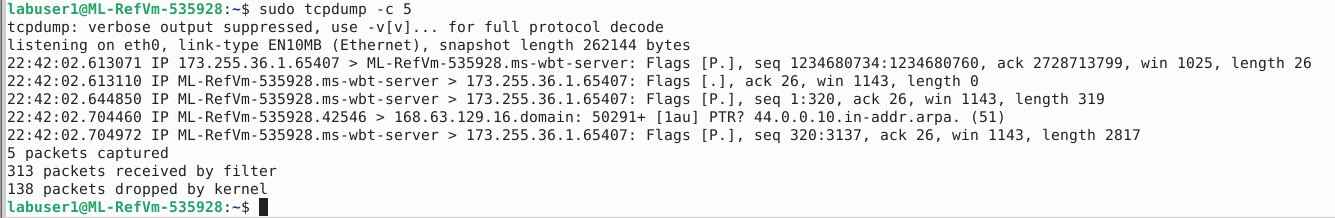
1. Display all open ports.\*



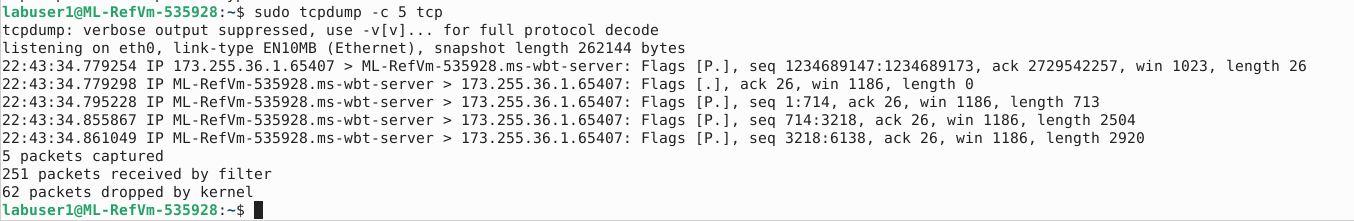
1. Display network traffic.\*



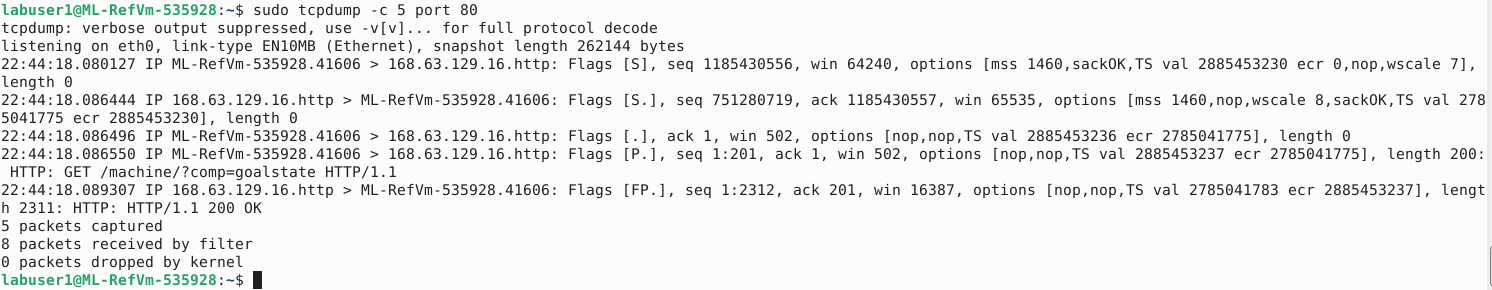
1. Display 5 packets of network traffic on eth0.\*



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



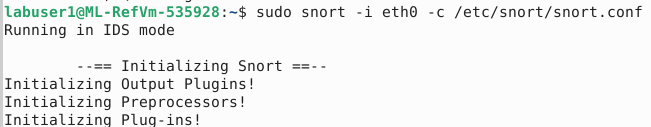
1. Display 5 packets of network traffic on port 80 and eth0.\*



Snort

<https://docs.snort.org/welcome>

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





1. Navigate to /etc/snort/rules/local.rules

Add rule:

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

1. Test the newly updated local.rules with:

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

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

