# Laboratory Exercise 4-2 – Scanning and Enumeration

## 1. Overview

For this lesson, students will use the Cyber Range: Kali Linux with Metasploitable3 Environment to perform Banner Grabbing with several different scanning tools. Students will also use Netstat to discover what ports are being listened to.
S_

## 2. Resources Required

This exercise requires a Kali Linux with Metasploitable3 Environment running in the Cyber Range.

## 3. Initial Setup

For this exercise, you will log in to your Cyber Range account and select the Kali Linux with Metasploitable3 Environment, then click "start" to start your environment and "join" to get to your Linux desktop login. Log in using these credentials:

Username: **`student`**
Password: **`student`**

## 4. Tasks

### Task 1: Banner Grabbing with Nmap

Remember that to run Metasploit you have to be root. The target we are attacking is still the Metasploitable machine (from the last lab exercise) as this is the only target in scope. The Metasploit IP will be denoted as <target IP>. **Please do not attack other IPs** as AWS will have several IPs that are out of scope but can be enumerated. Note that banner grabbing can work on several ports and services 80 (http), 21(FTP), 22 (SSH), 25 (SMTP), 23 (Telnet), 8080 (HTTP), and more. It will take practice to know when to use banner grabbing.

Open a terminal and complete the following commands:

1. Type **`service postgresql start`** and hit enter.
2. Type **`msfconsole`** and hit enter.
3. Type **`db_nmap -sV --script=banner <target IP>`** and hit enter.

```
msf6 > db_nmap -sV --script=banner 10.1.130.245
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-27 18:10 UTC
[*] Nmap: Nmap scan report for ip-10-1-130-245.ec2.internal (10.1.130.245)
[*] Nmap: Host is up (0.00036s latency).
[*] Nmap: Not shown: 991 filtered tcp ports (no-response)
[*] Nmap: PORT     STATE  SERVICE     VERSION
[*] Nmap: 21/tcp   open   ftp         ProFTPD 1.3.5
[*] Nmap: | banner: 220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.
1.1
[*] Nmap: |_30.245]
[*] Nmap: 22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Li
nux; protocol 2.0)
[*] Nmap: |_banner: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2
[*] Nmap: 80/tcp   open   http        Apache httpd 2.4.7
[*] Nmap: |_http-server-header: Apache/2.4.7 (Ubuntu)
[*] Nmap: 445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP
)
[*] Nmap: 631/tcp  open   ipp         CUPS 1.7
[*] Nmap: |_http-server-header: CUPS/1.7 IPP/2.1
[*] Nmap: 3000/tcp closed ppp
[*] Nmap: 3306/tcp open   mysql       MySQL (unauthorized)
[*] Nmap: | banner: U\x00\x00\x00\xFFj\x04Host 'ip-10-1-140-193.ec2.internal' is
 no
```

```
Mahesh - root@kali.example.com: /home/student/Desktop
File  Edit  View  Terminal  Tabs  Help
       =[ metasploit v6.4.9-dev                           ]
+ -- --=[ 2420 exploits - 1248 auxiliary - 423 post       ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > db_nmap -sV --script=banner 10.1.75.185
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-17 21:44 UTC
[*] Nmap: Nmap scan report for ip-10-1-75-185.ec2.internal (10.1.75.185)
[*] Nmap: Host is up (0.0000060s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT     STATE SERVICE       VERSION
[*] Nmap: 22/tcp   open  ssh           OpenSSH 9.7p1 Debian 5 (protocol 2.0)
[*] Nmap: |_banner: SSH-2.0-OpenSSH_9.7p1 Debian-5
[*] Nmap: 3389/tcp open  ms-wbt-server xrdp
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 21.67 seconds
msf6 >
```

Take a screenshot of the results and name it 4nmapbanner1. Save the scan in the scanning folder that you created for the previous lesson.

Type the following command:

**db_nmap -Pn -p 80,8484,8585,9200,139,137 -sV --script=banner <target IP>**

and hit enter.

As you can see, this scan specifies a few ports that we knew were on the target machine (recall from the previous lesson) but are not discovered with other scans. Because we are using the -Pn you may see filtered or closed. This can be inaccurate with this type of scan. The key is to discover what service is

running on the port. You can use the **`services`** command to view more details that are useful when researching vulnerabilities.

Take a screenshot of the results and name it 5nmapbanner2. Save the scan in the scanning folder.

```
msf6 > db_nmap -Pn -p 80,8484,8585,9200,139,137 -sV --script=banner 10.1.130.245
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-27 18:14 UTC
[*] Nmap: Nmap scan report for ip-10-1-130-245.ec2.internal (10.1.130.245)
[*] Nmap: Host is up (0.00013s latency).
[*] Nmap: PORT     STATE    SERVICE     VERSION
[*] Nmap: 80/tcp   open     http        Apache httpd 2.4.7
[*] Nmap: |_http-server-header: Apache/2.4.7 (Ubuntu)
[*] Nmap: 137/tcp  filtered netbios-ns
[*] Nmap: 139/tcp  filtered netbios-ssn
[*] Nmap: 8484/tcp filtered unknown
[*] Nmap: 8585/tcp filtered unknown
[*] Nmap: 9200/tcp filtered wap-wsp
[*] Nmap: MAC Address: 0A:FF:C7:3A:76:25 (Unknown)
[*] Nmap: Service Info: Host: target.example.com
[*] Nmap: Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 17.81 seconds
msf6 >
```

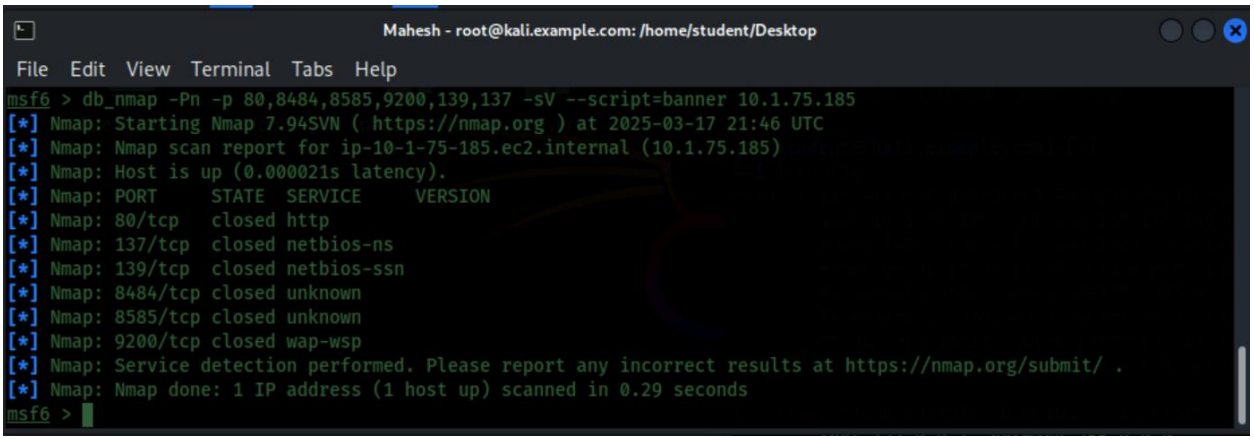
```
Mahesh - root@kali.example.com: /home/student/Desktop
File  Edit  View  Terminal  Tabs  Help
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
msf6 > services
Services
========

host         port  proto  name          state   info
----         ----  -----  ----          -----   ----
10.1.75.185  22    tcp    ssh           open    OpenSSH 9.7p1 Debian 5 protocol 2.0
10.1.75.185  80    tcp    http          closed
10.1.75.185  137   tcp    netbios-ns    closed
10.1.75.185  139   tcp    netbios-ssn   closed
10.1.75.185  3389  tcp    ms-wbt-server open    xrdp
10.1.75.185  8484  tcp                  closed
10.1.75.185  8585  tcp                  closed
10.1.75.185  9200  tcp    wap-wsp       closed

msf6 >
```

```
msf6 > services
Services
========

host         port  proto  name         state     info
----         ----  -----  ----         -----     ----
10.1.130.24  21    tcp    ftp          open      ProFTPD 1.3.5
5
10.1.130.24  22    tcp    ssh          open      OpenSSH 6.6.1p1 Ubuntu 2ubunt
5                                                u2 Ubuntu Linux; protocol 2.0
10.1.130.24  80    tcp    http         open      Apache httpd 2.4.7
5
10.1.130.24  137   tcp    netbios-ns   filtered
5
10.1.130.24  139   tcp    netbios-ssn  filtered
5
10.1.130.24  445   tcp    netbios-ssn  open      Samba smbd 3.X - 4.X workgrou
5                                                p: WORKGROUP
```




**Task 2: Banner Grabbing with cURL**

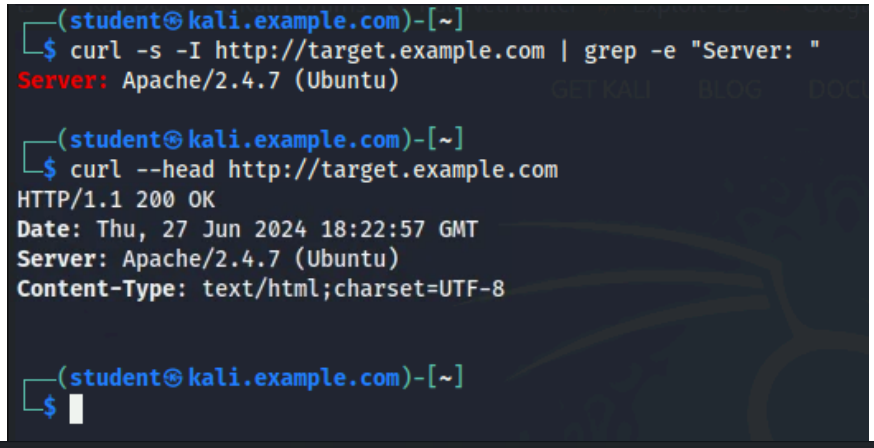
To use cURL we are going to need the domain. In this case, that is http://target.example.com. cURL will work in the msf console; however, it is best to execute this command in a new terminal, so that it can be easily referred back to.

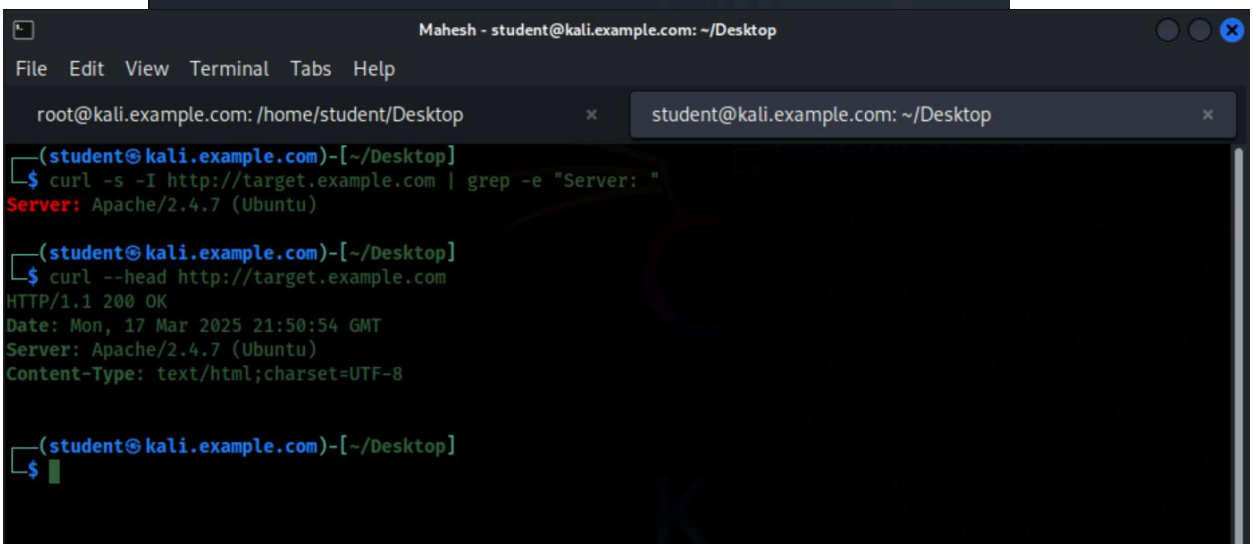
Open a new terminal tab (**File > Open Tab**) and complete the following commands:

1. Type **`curl -s -I http://target.example.com | grep -e "Server: "`** and hit enter.

2. Take a screenshot of the results and name it 6cURLbanner1 and save it to the scanning folder. These screenshots will be referenced in the report in later modules.

3. Type **`curl --head http://target.example.com`** and hit enter.



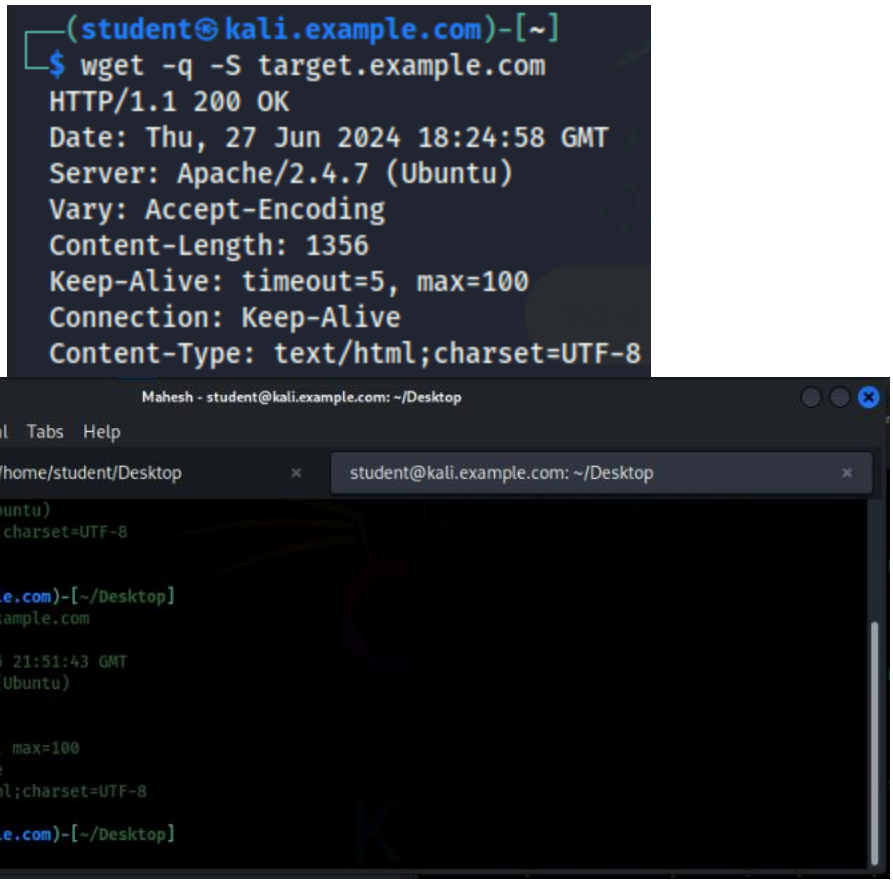




Take a screenshot of the results and name it 7cURLbanner2. Save the scan in the scanning folder.

**Task 3: Banner Grabbing with Wget**

Again, it is best to execute this command in a new terminal so it can be easily referred back to later. Take a screenshot of the results name it 8wgetbanner and save it to the scanning folder.

Type the following command: **`nc`** and hit enter.




**Task 4: Banner Grabbing with Netcat**

It is best to execute this command in a new terminal so it can be easily referred back to later. Remember to take a screenshot of the results and name it 9netcatbanner and save it to the scanning folder.

Type the following commands:

- **`nc -h`** and hit enter.

We do not want to resolve DNS, so we need a **`-n`**, and we want to print more results to the screen, so we will use **`-v`**. Combine them for a **`-nv`**.

- **`nc -nv <target IP> 22`** and hit enter. Press **`ctrl+c`** as soon as you see the server results.
- **`nc -nv <target IP> 21`** and hit enter. Press **`ctrl+c`** as soon as you see the server

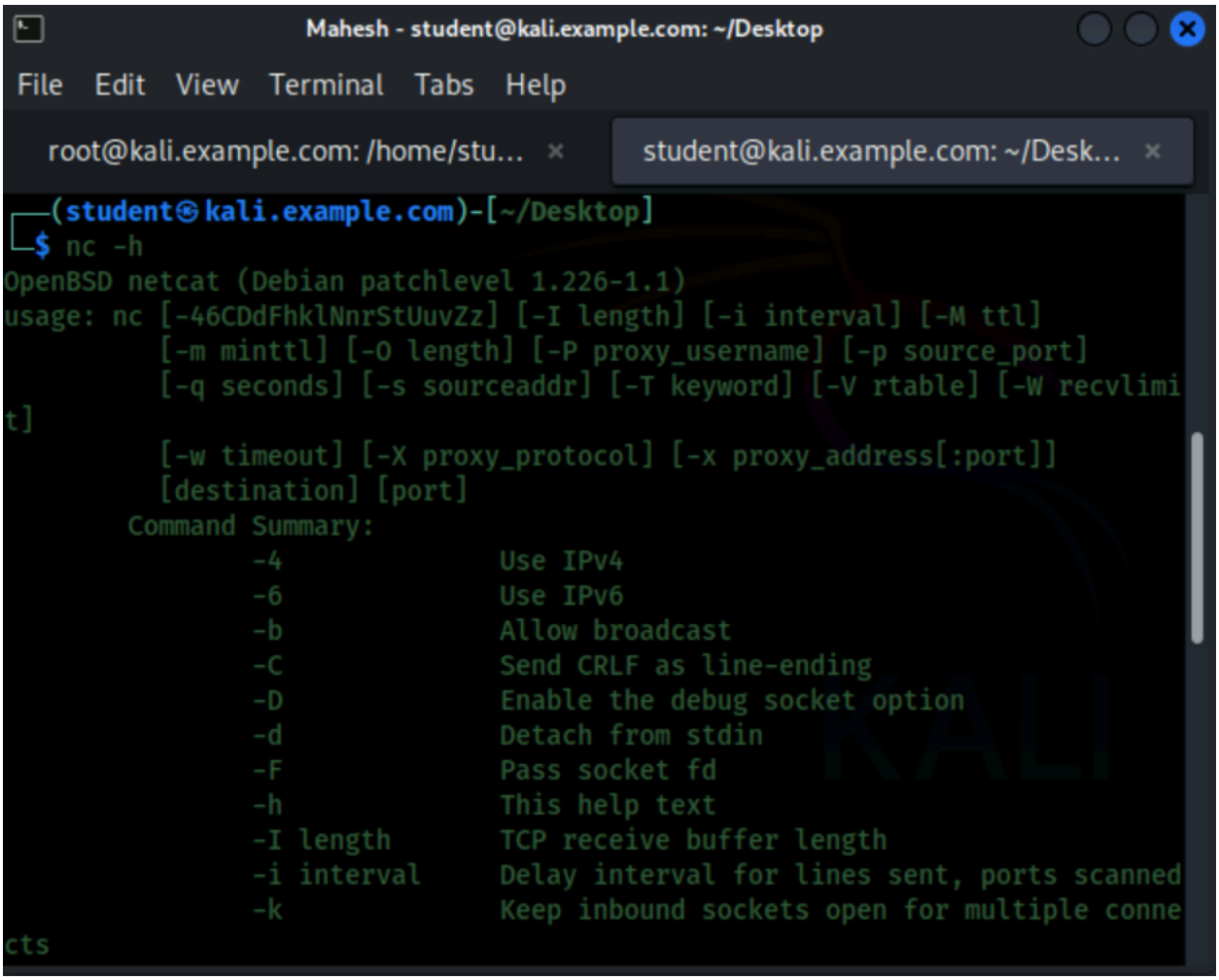
```
┌──(student㉿kali.example.com)-[~/Desktop]
└─$ nc -nv 10.1.75.185 22
Connection to 10.1.75.185 22 port [tcp/*] succeeded!
SSH-2.0-OpenSSH_9.7p1 Debian-5
^C

┌──(student㉿kali.example.com)-[~/Desktop]
└─$
```

```
┌──(student㉿kali.example.com)-[~]
└─$ nc -h
OpenBSD netcat (Debian patchlevel 1.226-1.1)
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
        Command Summary:
                -4              Use IPv4
                -6              Use IPv6
                -b              Allow broadcast
                -C              Send CRLF as line-ending
                -D              Enable the debug socket option
                -d              Detach from stdin
                -F              Pass socket fd
                -h              This help text
                -I length       TCP receive buffer length
                -i interval     Delay interval for lines sent, ports scanned
                -k              Keep inbound sockets open for multiple connects
                -l              Listen mode, for inbound connects
                -M ttl          Outgoing TTL / Hop Limit
                -m minttl       Minimum incoming TTL / Hop Limit
                -N              Shutdown the network socket after EOF on stdin
┌──(student㉿kali.example.com)-[~]
└─$ nc -nv 10.1.130.245 22
Connection to 10.1.130.245 22 port [tcp/*] succeeded!
SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2
^C

┌──(student㉿kali.example.com)-[~]
└─$ nc -nv 10.1.130.245 21
Connection to 10.1.130.245 21 port [tcp/*] succeeded!
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.1.130.245]
^C

┌──(student㉿kali.example.com)-[~]
└─$
```




**Task 5: Discovering listening ports with Netstat**

Remember that Netstat is a utility that will list network connections. Attackers can use this information to gain a better understanding of the network. It is best to execute this command in a new terminal so it can be easily referred back to later.

Type the following command: **`netstat -l`** and hit enter. This scan will list all listening ports. This is too much information and we need to narrow it down. Ensure you take a screenshot of the results name it 10netstat1 and save it to the scanning folder.

```
┌──(student㉿kali.example.com)-[~]
└─$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:postgresql    0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 localhost:postgresql    [::]:*                  LISTEN
tcp6       0      0 localhost:3350          [::]:*                  LISTEN
tcp6       0      0 [::]:ms-wbt-server      [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp6       0      0 fe80::8ff:dhcpv6-client [::]:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     8301     /run/xrdp/sockdir/xrd
p_chansrv_audio_out_socket_10
unix  2      [ ACC ]     STREAM     LISTENING     8302     /run/xrdp/sockdir/xrd
p_chansrv_audio_in_socket_10
unix  2      [ ACC ]     STREAM     LISTENING     863      /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     865      /run/systemd/userdb/i
o.systemd.DynamicUser
```

```
Mahesh - student@kali.example.com: ~/Desktop
File  Edit  View  Terminal  Tabs  Help
root@kali.example.com: /home/student/Desk...   student@kali.example.com: ~/Desktop
┌──(student㉿kali.example.com)-[~/Desktop]
└─$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:postgresql    0.0.0.0:*               LISTEN
tcp6       0      0 localhost:postgresql    [::]:*                  LISTEN
tcp6       0      0 localhost:3350          [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:ms-wbt-server      [::]:*                  LISTEN
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp6       0      0 fe80::c9a:dhcpv6-client [::]:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     9338     /run/xrdp/sockdir/xrdp_chansrv_audio_out_socket_10
unix  2      [ ACC ]     STREAM     LISTENING     456      /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     9339     /run/xrdp/sockdir/xrdp_chansrv_audio_in_socket_10
unix  2      [ ACC ]     STREAM     LISTENING     458      /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     459      /run/systemd/io.systemd.ManagedOOM
unix  2      [ ACC ]     STREAM     LISTENING     2111     /run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     12596    /var/run/postgresql/.s.PGSQL.5432
unix  2      [ ACC ]     STREAM     LISTENING     2117     /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     2119     /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     4466     /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     4468     /run/pcscd/pcscd.comm
unix  2      [ ACC ]     STREAM     LISTENING     5536     /tmp/.X11-unix/X10
unix  2      [ ACC ]     STREAM     LISTENING     556      /run/systemd/journal/io.systemd.journal
unix  2      [ ACC ]     STREAM     LISTENING     5452     /run/user/1000/systemd/private
```

Next, we will scan for only UDP ports that are listening. Type the following command:

**`netstat -l -u`** and hit enter.

```
┌──(student㉿kali.example.com)-[~]
└─$ netstat -l -u
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp6       0      0 fe80::8ff:dhcpv6-client [::]:*
```

```
┌──(student㉿kali.example.com)-[~/Desktop]
└─$ netstat -l -u
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp6       0      0 fe80::c9a:dhcpv6-client [::]:*

┌──(student㉿kali.example.com)-[~/Desktop]
```

You can filter the scan with grep by port or service. Type the following command: **`netstat -l | grep ssh`** and hit enter.

```
┌──(student㉿kali.example.com)-[~]
└─$ netstat -l | grep ssh
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
unix  2      [ ACC ]     STREAM     LISTENING     6520     /tmp/ssh-739RQ0ReDBMz
/agent.1134
unix  2      [ ACC ]     STREAM     LISTENING     5816     /run/user/1000/gcr/ss
h
unix  2      [ ACC ]     STREAM     LISTENING     6127     /run/user/0/gcr/ssh
unix  2      [ ACC ]     STREAM     LISTENING     5824     /run/user/1000/gnupg/
S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     6135     /run/user/0/gnupg/S.g
pg-agent.ssh
```

```
┌──(student㉿kali.example.com)-[~/Desktop]
└─$ netstat -l | grep ssh
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
unix  2      [ ACC ]     STREAM     LISTENING     6617     /run/user/1000/gcr/ssh
unix  2      [ ACC ]     STREAM     LISTENING     6913     /run/user/0/gcr/ssh
unix  2      [ ACC ]     STREAM     LISTENING     6625     /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     6921     /run/user/0/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     5956     /tmp/ssh-PHEpHi1lh9FG/agent.1185

┌──(student㉿kali.example.com)-[~/Desktop]
└─$
```

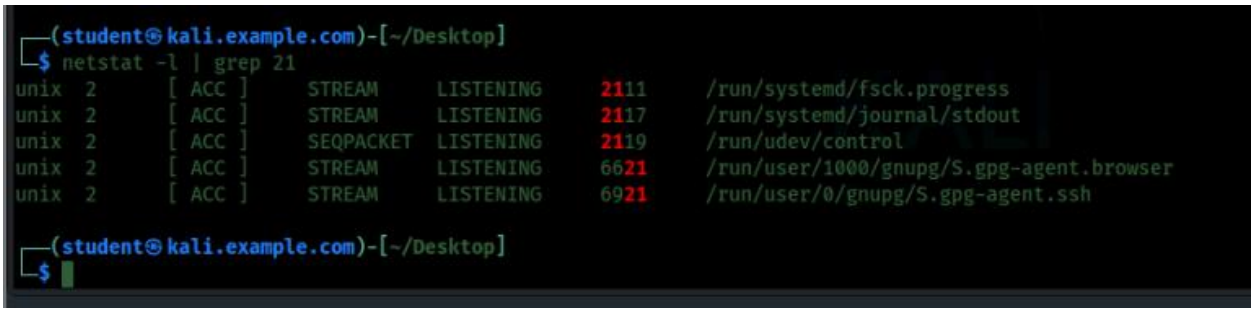
Now let's try a few more ways to filter. Type the following commands:

- **`netstat -l | grep rdp`** and hit enter.

```
┌──(student㉿kali.example.com)-[~/Desktop]
└─$ netstat -l | grep rdp
unix  2      [ ACC ]     STREAM     LISTENING     9338     /run/xrdp/sockdir/xrdp_chansrv_audio_out_socket_10
unix  2      [ ACC ]     STREAM     LISTENING     9339     /run/xrdp/sockdir/xrdp_chansrv_audio_in_socket_10
unix  2      [ ACC ]     STREAM     LISTENING     5696     /run/xrdp/sockdir/xrdpapi_10
unix  2      [ ACC ]     STREAM     LISTENING     5716     /run/xrdp/sockdir/xrdp_display_10

┌──(student㉿kali.example.com)-[~/Desktop]
└─$
```

- and hit enter.

- 


What we are attempting to do is gain as much information about the network as possible so that we can find vulnerabilities. Netstat is useful to administrators as a check to see if attackers have opened or connected to listening ports; however, the Netstat tool will also allow an attacker to see all listening ports. This will provide an attacker with a further understanding of the network and what services they may be able to exploit.

**5. References:**

https://www.aelius.com/njh/subnet_sheet.html
https://nmap.org/book/nse-usage.html