Lab Assignment 8

Aim: Installation of NMAP and using it with different options to scan open ports, perform OS fingerprinting, do a ping scan, tcp port scan, udp port scan, etc.

Lab Outcome Attainment: LO4

A port scan is a series of messages sent by someone attempting to break into a computer to learn which computer network services, each associated with a "well-known" port number, the computer provides. Port scanning, a favourite approach of computer cracker, gives the assailant an idea where to probe for weaknesses. Essentially, a port scan consists of sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can therefore be probed for weakness.

Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts.

Nmap uses raw IP packets in novel ways to determine

- what hosts are available on the network,
- what services (application name and version) those hosts are offering,
- what operating systems (and OS versions) they are running,
- what type of packet filters/firewalls are in use etc.

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the "interesting ports table". **That table lists the port number and protocol, service name, and state.**

The state is either open, filtered, closed, or unfiltered.

- Open means that an application on the target machine is listening for connections/packets on that port.
- **Filtered** means that a firewall, filter, or other network obstacle is blocking the port so that Nmap cannot tell whether it

is open or closed.

- **Closed** ports have no application listening on them, though they could open up at any time.
- Ports are classified as **unfiltered** when they are responsive to Nmap's probes, but Nmap cannot determine whether they are open or closed.
- Nmap reports the state combinations **open|filtered** and **closed|filtered** when it cannot determine which of the two states describe a port.

Nmap for typical scan:

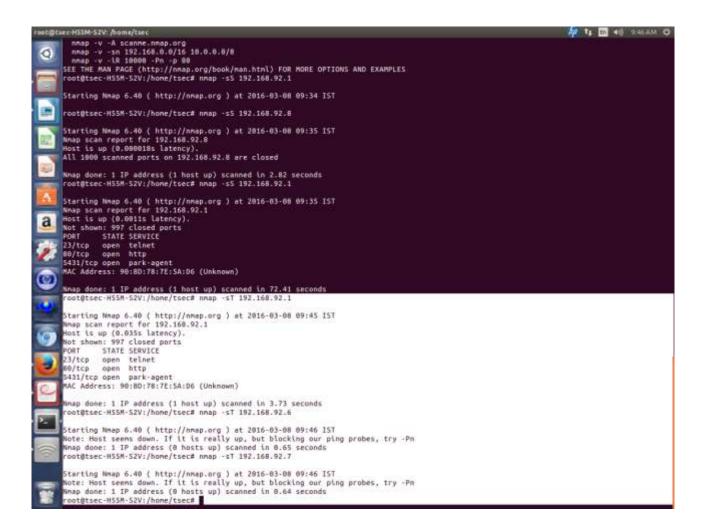
1. TCP connect(0) scanning# nmap -sT ipadress

These scans are so called because UNIX sockets programming uses a system call named connect() to begin a TCP connection to a remote site. If connect() succeeds, a connection was made. If it fails, the connection could not be made (the remote system is offline, the port is closed, or some other error occurred along the way). This allows a basic type of port scan, which attempts to connect to every port in turn, and notes whether or not the connection succeeded. Once the scan is completed, ports to which a connection could be established are listed as open, the rest are said to be closed.

This method of scanning is very effective, and provides a clear picture of the ports you can and cannot access. If a connect() scan lists a port as open, you can definitely connect to it - that is what the scanning computer just did! There is, however, a major drawback to this kind of scan; it is very easy to detect on the system being scanned. If a firewall or intrusion detection system is running on the victim, attempts to connect() to every port on the system will almost always trigger a warning. Indeed, with

modern firewalls, an attempt to connect to a single port which has been blocked or has not been specifically "opened" will usually result in the connection attempt being logged. Additionally, most servers will log connections and their source IP, so it would be easy to detect the source of a TCP connect() scan.

For this reason, the TCP Stealth Scan was developed.



2. TCP SYN scan

#nmap -sS ipadress

SYN or Stealth scanning makes use of this procedure by sending a SYN packet and looking at the response. If SYN/ACK is sent back, the port is open and the remote end is trying to open a TCP connection. The scanner then sends an RST to tear down the connection before it can be established fully; often preventing the connection attempt appearing in application logs. If the port is closed, an RST will be sent. If it is filtered, the SYN packet will have been dropped and no response will be sent. In this way, Nmap can detect three port states - open, closed and filtered. Filtered ports may require further probing since they could be subject to firewall rules which render them open to some IPs or conditions, and closed to others.

Modern firewalls and Intrusion Detection Systems can detect SYN scans, but in combination with other features of Nmap, it is possible to create a virtually undetectable SYN scan by altering timing and other options.

```
reactions with the common control in the three major formats at once

v. Increase verbosity level (use v.v or nore for greater effect)

-d. Increase verbosity level (use v.v or nore for greater effect)

-d. Increase verbosity level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use v.v or nore for greater effect)

-d. Increase debugging level (use
```

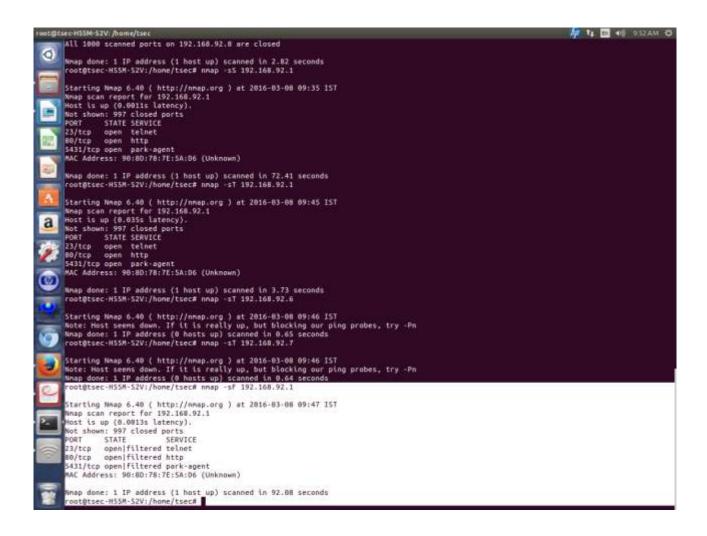
3. TCP FIN scan

#nmap -sF ipaddress

The idea behind these type of scans is that a closed port should respond with an RST upon receiving packets, whereas an open port should just drop them (it's listening for packets with SYN set). This way, you never make even part of a connection, and never send a SYN packet; which is what most IDS' look out for.

The FIN scan sends a packet with only the FIN flag set.

These scan types will work against any system where the TCP/IP implementation follows RFC 793. Microsoft Windows does not follow the RFC, and will ignore these packets even on closed ports. This technicality allows you to detect an MS Windows system by running SYN along with one of these scans. If the SYN scan shows open ports, and the FIN/NUL/XMAS does not, chances are you're looking at a Windows box



4. TCP NULL scan

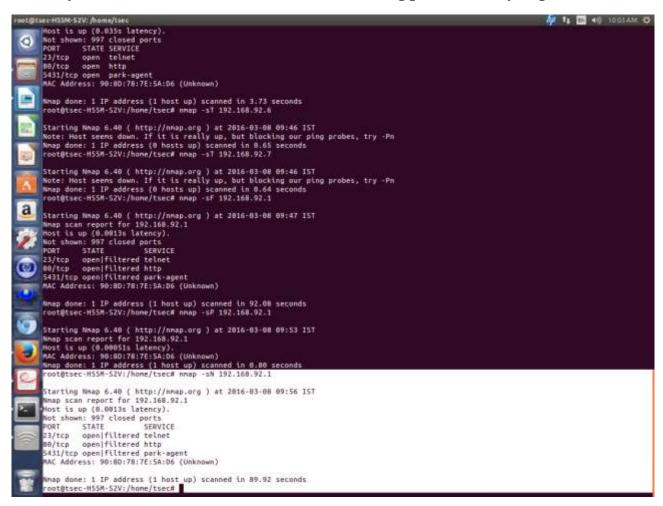
The Null Scan is a type of TCP scan that hackers — both ethical and malicious — use to identify listening TCP ports. In the right hands, a Null Scan can help identify potential holes for server hardening, but in the wrong hands, it is a reconnaissance tool. It is a pre-attack probe.

A Null Scan is a series of TCP packets that contain a sequence number of 0 and no set flags. In a production environment, there will never be a TCP packet that doesn't contain a flag. Because the Null Scan does not contain any set flags, it can sometimes penetrate firewalls and edge routers that filter incoming packets with particular flags.

The expected result of a Null Scan on an open port is no response. Since there are no flags set, the target will not know how to handle the request. It will discard the packet and no reply will be sent. If the port

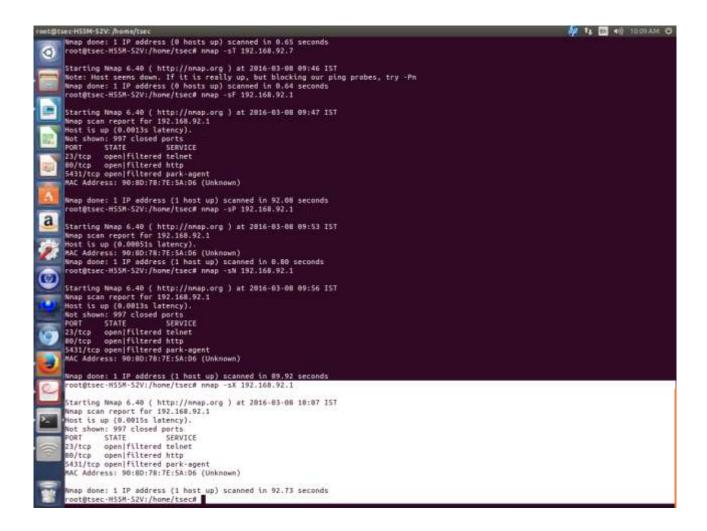
is closed, the target will send an RST packet in response.

Information about which ports are open can be useful to hackers, as it will identify active devices and their TCP-based application-layer protocol.



5. XMAS scan

The Xmas Tree scan sets the FIN, URG and PUSH flags are set. This scan will work on UNIX and related systems and cause the kernel to drop the packet if the receiving port is open.



6. Ping sweep

nmap -sP IP address of gateway

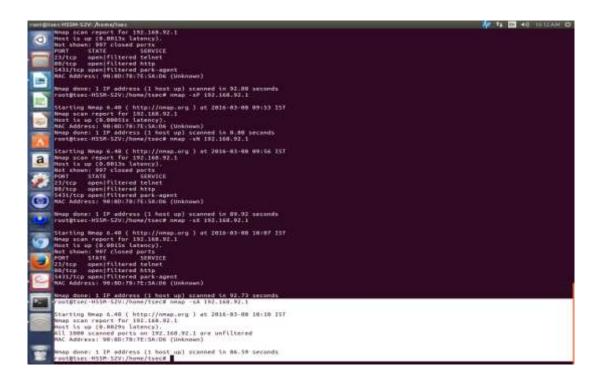
This scan type lists the hosts within the specified range that responded to a ping. It allows you to detect which computers are online, rather than which ports are open.

7. ACK scan

This scan type sends ACK packets to a host.

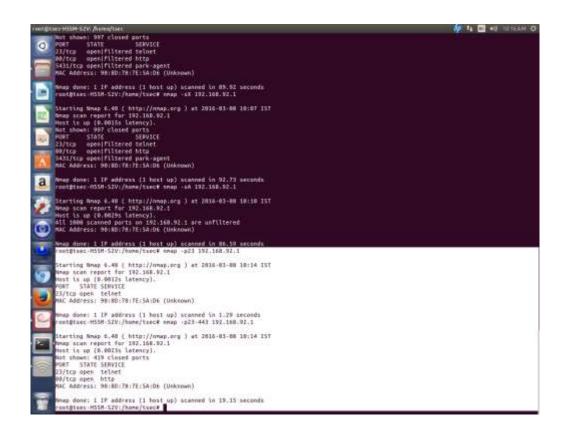
If an RST comes back, the port is classified "unfiltered" (that is, it was allowed to send its RST through whatever firewall was in place). If nothing comes back, the port is said to be "filtered", that is, the firewall prevented the RST coming back from the port. This scan type can help determine if a firewall is stateless (just blocks incoming SYN packets) or stateful (tracks connections and also blocks unsolicited ACK packets).

Note that an ACK scan will never show ports in the "open" state, and so it should be used in conjunction with another scan type to gain more information about firewalls or packet filters between yourself and the victim.



8. scanning range of ports

#nmap -p23 ipaddress scans specific port
#nmap -p23-443 ipaddress scans ports ranging from 23 to 443

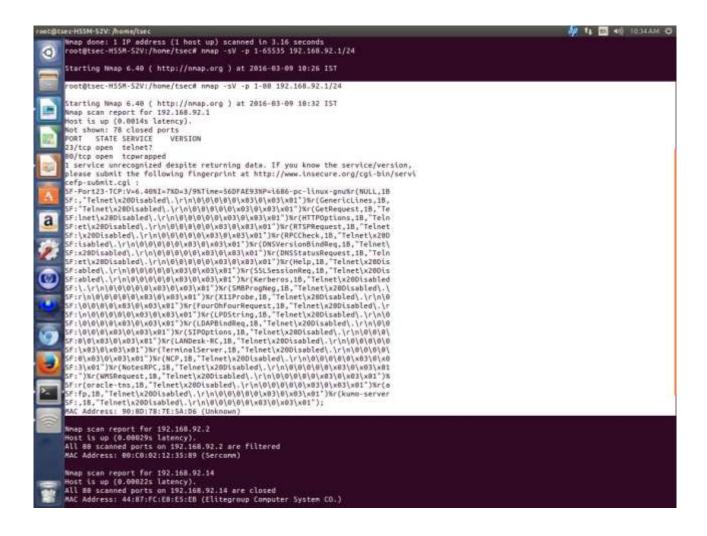


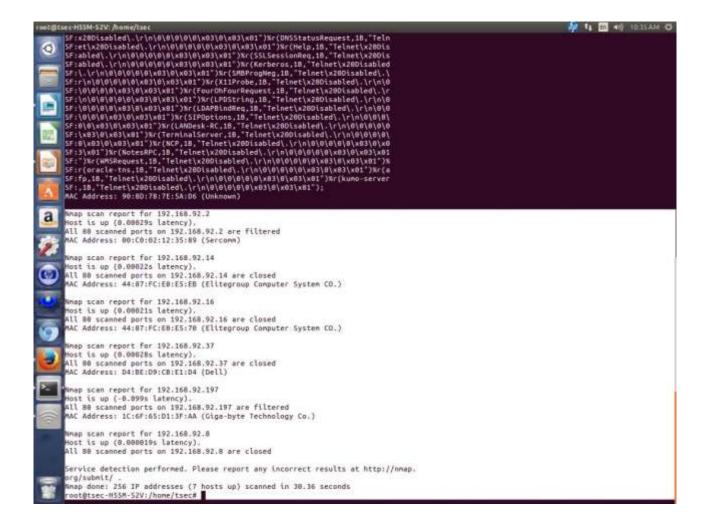
9. Version Detection [-sV]

Version Detection collects information about the specific service running on an open port, including the product name and version number. This information can be critical in determining an entry point for an attack. The -sV option enables version detection, and the -A option enables both OS fingerprinting and version detection, as well as any other advanced features which may be added in future releases.

10. nmap -sV -p 1-80 192.168.1.1/24

This command will scan all of your local IP range (assuming your in the 192.168.1.0-254 range), and will perform service identification (-sV) and will scan all ports (-p 1-80). Since you are running this as a normal user and not root it will be TCP Connect based scan. If you run the command with sudo at the front it will run as a TCP SYN scan.

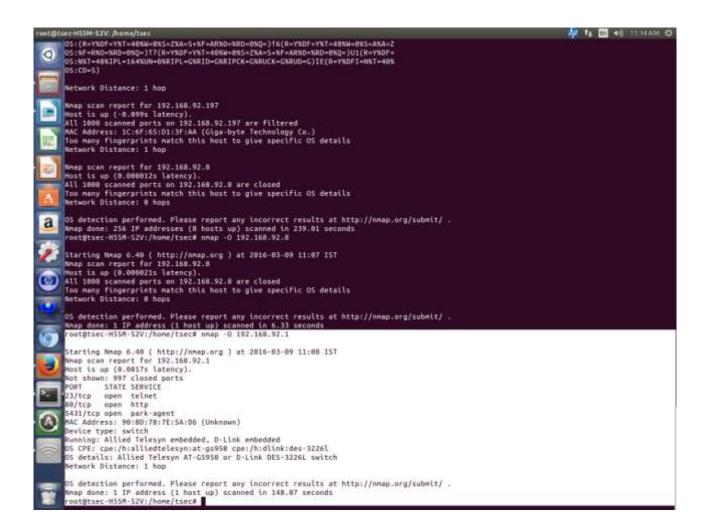




11. Operating system fingerprining

#nmap -O ipaddress

Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After performing dozens of tests such as TCP ISN sampling, TCP options support and ordering, IP ID sampling, and the initial window size check, Nmap compares the results to its nmap-os-db database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match.



12. Idle scan - self study