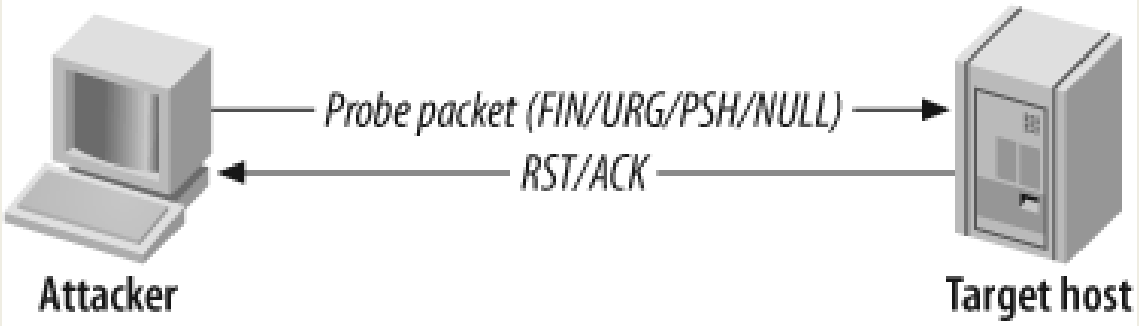
SCANNING NETWORK

Bappe Sarker

What is Network Scan?

Scanning is typically an automated process that is used to discover devices such as PC, server and peripherals that exist on a network. Results can include details of the discovered devices, including **IP addresses** , device names, operating systems, running applications/services, open shares, usernames and groups. Scanning is often related to pre - attack or reconnaissance activities.



Network Scanning Tools

- Nmap / Zenmap
- Hping2 / Hping3
- Masscan
- Angy Ip Scanner
- Netdiscover
- Rastscan

Understand TCP Flags

- SYN bit is used in the initial three-way handshake where both parties generate the initial sequence numbers.
- ACK is used to confirm that the data packets have been received.
- FIN bit is used to end the TCP connection. TCP is a full-duplex, so both the sender and receiver must use the FIN bit to end the connection. This is the standard method of how both parties end the connection.
- RST resets the connection. When the host receives this, it must terminate the connection right away. This is only used when there are unrecoverable errors, and it is not a normal way to finish the TCP connection.
- URG says that the data should be treated with priority over other data.
- PSH tells an application that the data should be transmitted immediately, and we do not want to wait to fill the entire TCP segment.

Discovery Scan

- Network discovery scanning is **the first step in a security assessment of a system**. Network Discovery Scans scan a range of IP addresses, searching for nodes those are alive.

- nmap –sn -PR <target_ip> [-sn = disable port scan, -PR = ARP ping scan]

Common scanning Techniques

- **TCP scan** (complete scan with tcp three way handshake)

namp –sT <target_ip>

- **UDP Scan** (scan UDP Protcol)

namp –sU <target_ip>

- **SYN Scan / Stealth scan** (Scan same as TCP scan but required less step)

namp –sS <target_ip>

Nmap Common Parameters for scanning

- Port Range

namp -p [80-100 / 80,445,449 / 0-65535 / -p-]

- Service Version

nmap –sV

- OS information

nmap –O

- Verbose level

namp –v / -vv / -vvv

Nmap Scan Speed

- In production environment, there are various types of security controls are implemented for that reason they can nmap as noisy traffic. To control the speed we can use –T parameter

- Nmap –T0/T1/T2/T3/T4/T5 (min to max)

Inverse Scan To bypass firewall

- Inverse scan is fully opposite of TCP or regular scan. That's mean the initial flag is RST / FIN / NULL . There are few types of inverse scan available but all are same provides same output.

- Xmass Scan

namp –sX <target_ip>

- Maimon scan

namp –sM <target_ip>

Scan Behind Firewall

- **Packet fragmentation scan**
- # namp –sT <target_ip>
- **Decoy Scan**

namp –D RND:10 <target_ip>

- **Source port manipulation scan**

namp –sg 80 -p 445 <target_ip>

Nmap scripts

- The Nmap Scripting Engine (NSE) extends Nmap's capabilities to enable it to perform a variety of tasks and report the results along with Nmap's normal output.

#locate .nse (locate the nse file)

#nmap –p 445 - - sV –script=smb-vuln-ms17-010.nse <target_ip>

Nmap Automator

- The main goal for this script is to automate the process of enumeration & recon that is run every time, and instead focus our attention on real pentesting.
- This will ensure two things:
- Automate nmap scans.
- Always have some recon running in the background.
- Once initial ports are found '*in 5-10 seconds*', we can start manually looking into those ports, and let the rest run in the background with no interaction from our side whatsoever.

Rustscan – The fastest scanning tool

- **RustScan** is the tool that assures the fastest result retrieving tool as compares to Nmap. RustScan is a tool that turns a 17 minutes Nmap scan into 19 seconds. RustScan tool is developed in the Rust language and valid on the GitHub platform. RustScan tool is an open-source and free-to-use tool. RustScan tool can scan 65k ports in almost 7-8 seconds which is much faster than other tools. RustScan tool has support to IPv6 Version IP.

Example commands

rustscan --ip <target_ip> --ports <port_range> --tcp

Thank You