SECURITY WALKTHROUGH <Nmap>

Submitted to: << sprints >>

Security Analyst: << Ali Mohamed Abdelfatah >>
Security Analyst: << Mohamed Ahmed Fathy>>
Security Analyst: << Tarek Ayman Hassan>>
Security Analyst: << Ali Samy Gomaa>>
Security Analyst: << Zyad Mohamed Hagag>>

When port scanning with Nmap, there are three basic scan types. These are:

- TCP Connect Scans (-sT)

  It performs a full **TCP three-way handshake** with the target (SYN → SYN-ACK → ACK).
  If the handshake is successful, the port is reported as **open**.
  If the target responds with a **RST** packet, the port is considered **closed**.

```
nmap -sT <target_ip>
```

- SYN "Half-open" Scans (-sS)

1. Nmap sends a **SYN** packet to a port.
2. If the target responds with a **SYN-ACK**, the port is considered **open**.
3. Nmap then sends a **RST** packet instead of completing the handshake, leaving the connection half-open.
4. If the target responds with a **RST**, the port is considered **closed**.

```
nmap -sS <target_ip>
```

- UDP Scans (-sU)

1. Nmap sends a **UDP packet** to a port.
2. If no response is received, the port is assumed to be **open or filtered**.
3. If an **ICMP unreachable** message is received, the port is considered **closed**.

```
nmap -sU <target_ip>
```

Additionally, there are several less common port scan types, some of which we will also cover (albeit in less detail). These are:

- TCP Null Scans (-sN)

1. If the port is **closed**, the target responds with a **RST** packet.

2. If the port is **open**, no response is sent (in most cases).

```
nmap -sN <target_ip>
```

- TCP FIN Scans (-sF)

1. If the port is **closed**, the target responds with a **RST** packet.
2. If the port is **open**, no response is sent.

```
nmap -sF <target_ip>
```

TCP Xmas Scans (-sX)

1. If the port is **closed**, the target responds with a **RST** packet.
2. If the port is **open**, no response is sent.

```
nmap -sX <target_ip>
```

UDP Scanning with Nmap

How UDP Scans Work

1. **Stateless Nature**: UDP doesn't involve a handshake (like TCP's three-way handshake). Instead, data is sent, and the sender hopes it arrives. There's no confirmation that the packet was received. This makes it hard to know if a UDP port is open or closed without a response.
2. **Handling Open and Closed Ports**:
    - **Open Ports**: When you send a packet to an open UDP port, **there is typically no response**. This creates uncertainty—Nmap can't be sure if the port is open or if the packet was dropped by a firewall. Nmap, therefore, marks these ports as **open|filtered**, meaning it could be open or filtered (blocked by a firewall).
    - **Closed Ports**: If the port is closed, the target should respond with an **ICMP "Port Unreachable"** message (Type 3, Code 3), which clearly identifies the port as **closed**.
3. **Scanning Process**:
    - Nmap sends a UDP packet to a target port. ○ If there's **no response**, Nmap suspects the port is **open or filtered**. ○ To confirm, Nmap sends the request a second time to double-check.
    - If there's still no response, Nmap marks the port as **open|filtered**. ○ If Nmap receives an **ICMP "Port Unreachable"** message, the port is marked as **closed**.

A basic UDP scan on a target would look like this:

```
nmap -sU <target_ip>
```

To scan the **top 20 UDP ports**:

```
nmap -sU --top-ports 20 <target_ip>
```

top ports UDP scan first to quickly gather information

```
nmap -sU --top-ports 50 <target_ip>
```

What is a Ping Sweep?

A **ping sweep** is a method to determine which hosts in a network are up and running by sending **ICMP echo requests** (commonly known as **ping**). For each IP address that responds, Nmap will mark the host as "alive" or active.

Nmap Ping Sweep with `-sn`

When using Nmap for a ping sweep, the `-sn` option disables port scanning, ensuring that Nmap focuses on host discovery only. It sends:

- **ICMP Echo Requests** (or ARP requests for local networks) • **TCP SYN to port 443** (HTTPS)
- **TCP ACK to port 80** (HTTP) if the scan is not run as root, it sends SYN packets instead.

Using a range of IP addresses:

```
nmap -sn 192.168.0.1-254
```

Using **CIDR notation**:

```
nmap -sn 192.168.0.0/24
```

Key NSE Script Categories

Here are some of the most useful categories of NSE scripts:

1. **safe**

These scripts are non-intrusive and won't harm the target in any way. They are safe to run in most scenarios, such as basic recon.

```
nmap --script=safe <target>
```

Scripts in this category are likely to affect the target. They may slow down services, cause crashes, or otherwise have side effects.

```
nmap --script=intrusive <target>
```

vuln

These scripts are used for vulnerability scanning. They identify known vulnerabilities in services running on the target.

```
nmap --script=vuln <target>
```

exploit

Exploit scripts attempt to actively exploit a vulnerability on the target system. These are generally used after a vulnerability is identified.

```
nmap --script=exploit <target>
```

auth

These scripts try to bypass or brute-force authentication mechanisms of various services.

```
nmap --script=auth <target>
```

brute

These scripts attempt to **brute-force** credentials for services, such as SSH, FTP, or HTTP basic auth.

```
nmap --script=brute <target>
```

discovery
This category is designed to extract additional information about a target. It can be used to query running services for more details about the network.

```
nmap --script=discovery <target>
```

To run an Nmap scan with a specific script or set of scripts, use the `--script` flag followed by the name of the script or category. For instance, to run a **vulnerability scan**:

```
nmap --script=vuln <target>
```

If you want to scan a specific vulnerability (like SMBv1):

```
nmap --script=smb-vuln-ms17-010.nse <target>
```

- NSE scripts are stored in the **Nmap scripts library** (`/usr/share/nmap/scripts/` on Linux).
- You can view detailed information about a script, including usage and arguments, by using the `–script-help` flag:

```
nmap --script-help <script_name>
```

Bypassing Firewalls with Nmap

1. **The -Pn Option: Ignoring Host Discovery**

When a host blocks **ICMP packets** (common with Windows firewalls), Nmap may incorrectly assume the host is down if it doesn't receive a ping reply. The `-Pn` option tells Nmap to skip host discovery and assume the host is alive.

```
nmap -Pn <target>
```

2. **Fragmenting Packets (-f Option)**

Some firewalls and Intrusion Detection Systems (IDS) look for specific patterns in large packets to block scans. The `-f` option breaks packets into smaller fragments, making it harder for a firewall or IDS to reassemble and detect the full packet.

```
nmap -f <target>
```

3. **Adding Delays Between Packets (--scan-delay)**

Some firewalls or IDS devices use **rate-limiting** or **timing thresholds** to detect and block rapid scanning attempts. By adding delays between packets, you can potentially evade detection.

```
nmap --scan-delay 100ms <target>
```

4. **Generating Bad Checksums (--badsum)**

The `--badsum` option sends packets with an **invalid checksum**, which means that any correctly configured TCP/IP stack will drop the packet. However, some firewalls/IDS may respond automatically to the packet, indicating their presence.

```
nmap --badsum <target>
```

5. **Using ARP for Local Networks**

On a **local network**, Nmap can use **ARP requests** to detect hosts, which is much more reliable than using ICMP ping requests. This method works even if ICMP or other forms of host discovery are blocked, as ARP is necessary for basic network communication on local subnets.

```
sudo nmap -sn 192.168.1.0/24
```