Lab 6 - Network Discovery and Scanning

Task 1 - Setting Up the Testing Environment (30 minutes)

This task ensures the correct setup of the virtual machines (VMs) and network configuration required for the lab. The following steps guide you through the process.

A **NAT** (Network Address Translation) network isolates the VMs from the external internet but allows them to communicate with each other and your host machine. It's useful for safely conducting network scans without affecting other devices on your physical network.

1. **Check whether a NAT network exists in VirtualBox:**
   Open VirtualBox and go to 'Preferences'.
   Select the 'Network' tab.
   If no NAT network exists, you can manually add a NAT network using the command line:
   Windows:

1. `cd "C:\Program Files\Oracle\VirtualBox"` **or in macOS/Linux:** `cd /usr/bin/VirtualBox`
2. `VBoxManage natnetwork add --netname NatNetwork --network "10.0.2.0/24" --enable --dhcp on`
3. `VBoxManage list natnetworks`

- Open VirtualBox and right-click your **Kali Linux VM**.
- Go to **Settings → Network**.
- Set the adapter to:

  - **Attached to**: NAT Network
  - **Name**: Select `NatNetwork` from the dropdown.

- Start the Kali VM.
  - Run `ifconfig` in the terminal to verify it has received an IP address in the range `10.0.2.x`. and write the output in the lab report

2. **Test network adapters on your VM:**
   Set Adapter 1 to 'Bridged Adapter', start the Kali Linux VM, and run the command:
   `ifconfig`

Record the IP address and subnet mask.
Shut down the VM, change Adapter 1 to 'NAT Network', and repeat the process.

Real-life analogy: Imagine setting up two separate neighborhoods (Bridged and NAT networks). In Bridged, all houses (VMs) connect to the same external world (your real network). In NAT, the houses are isolated, but they can still talk to each other within the neighborhood.

A **NAT** network acts like a "private neighborhood" where only devices (VMs) within the same network can communicate with each other.
The VMs are shielded from the outside world (e.g., your physical network or the internet), preventing unintended interference or risk

For lab tasks, VMs like **Kali Linux** (the attacker machine) must discover and interact with other VMs, such as **Metasploitable** (the target machine).
The NAT network ensures they can find each other and interact as if they are on a small, local area network (LAN)

Task 2 - Exploring the Network (20 minutes)

In Task 2, you will explore the network to identify devices (hosts), check their activity, and gather information using tools like **arp** and **nmap**. This task mimics how network administrators or security analysts identify active devices and assess the network's structure.

1. **Start the Metasploitable VM and ensure its network adapter is set to 'NAT Network'.**
   Log in to Metasploitable (*default username/password: msfadmin/msfadmin*).

   Metasploitable acts as the target system in this lab. It must be in the same NAT network as Kali Linux so they can communicate and be discovered during scans.

2. **Discover neighbors using the command:**

   ```
   arp -a
   ```

   This command lists devices on the network with their IP and MAC addresses. It's like walking around a neighborhood and noting which houses are occupied.
   Before this step, I suggest you try pinging the target machine.

The `arp -a` command checks which devices (neighbors) on the same subnet recently communicated with your machine.

Purpose: Provides a quick, passive method to identify devices within the local network.

3. **Perform a ping scan using nmap:**

```
nmap 10.0.2.0/24 -sn
```

This checks for active devices in the subnet without probing for detailed information. Think of this as knocking on every door in the neighborhood to see who answers.

The `nmap 10.0.2.0/24 -sn` command actively scans the network to locate all devices, including those that might not show up in `arp`.

Purpose: Complements `arp` by actively probing the network for all reachable devices, providing a more comprehensive view.

4. **Exclude specific addresses from the scan:**

```
nmap --exclude 10.0.2.1,10.0.2.3 10.0.2.0/24 -sn
```

If specific devices should not be scanned, run this command.
This ensures sensitive devices are not disturbed. Avoid ringing the doorbells of certain houses in the neighborhood.

The primary purpose of **Task 2** is to **discover and map the devices on a network**. It simulates the process of identifying all active devices (hosts) within a specific network, such as computers, servers, or other connected devices. This is a foundational step in both **network administration** and **cybersecurity**:

1. **Network Administration**:
   - Helps identify which devices are connected to the network and their activity.
   - Provides a clear picture of the network layout, which is essential for troubleshooting, maintenance, and planning.
2. **Cybersecurity**:
   - Simulates how an attacker might probe a network to locate potential targets.
   - It helps defenders understand how to secure their network by identifying open devices and misconfigurations.

Task 3 - OS and Service Detection (20 minutes)

In Task 3, the goal is to identify the operating systems (OS) and services running on devices in the network. This is a critical task in both network management and cybersecurity, as it provides detailed insights into the devices, their roles, and potential vulnerabilities.

1. **Detect operating systems:**

   ```
   sudo nmap -O 10.0.2.0/24
   ```

   This attempts to identify the OS of devices based on their responses to specific probes.

   - Sends special probes to devices to analyze their behavior.
   - Matches the responses against Nmap's OS fingerprint database.
   - Tries to guess the OS, kernel version, and additional details

   Each device in a network serves a specific purpose (e.g., a web server, database server, or file server). Knowing the OS helps identify its role. Certain operating systems or services are more vulnerable to specific attacks, especially if they're outdated or misconfigured.

   For example:

   - A device running Windows Server might be a domain controller.
   - A device running Linux might host a web application.
   - *Windows XP (still found in legacy systems) is highly vulnerable because it's no longer supported.*
   - *An outdated Apache HTTP server might have known exploits*

2. **Detect service versions:**

   ```
   sudo nmap -sV -n 10.0.2.0/24
   ```

   This command lists open ports and their services with version details.
   Real-life analogy: Identifying the brand and model of appliances in each house.

3. **Analyze results:**

   Record the output, focusing on FTP, HTTP, and database servers. Determine if any outdated or vulnerable versions are running.

Example Analysis of a target machine (Metasploitable):

Host: 10.0.2.4 (Metasploitable)

- **Details**:
  - **Open Ports**:
    - FTP (`21/tcp`) - vsftpd 2.3.4 (known vulnerabilities).
    - SSH (`22/tcp`) - OpenSSH 4.7p1 (outdated).
    - Telnet (`23/tcp`) - Linux Telnetd (insecure protocol).
    - HTTP (`80/tcp`) - Apache HTTPD 2.2.8 (outdated).
    - Database Services: MySQL (`3306/tcp`), PostgreSQL (`5432/tcp`).
    - Other notable services:
      - Samba file sharing (`139/tcp`, `445/tcp`).
      - NFS (`2049/tcp`) for network file sharing.
      - UnrealIRCd (`6667/tcp`), a vulnerable IRC server.
      - Apache Tomcat (`8180/tcp`), a JSP engine.
  - **MAC Address**: `08:00:27:16:FB:A6` (Oracle VirtualBox virtual NIC).
  - **Service Info**: Detected as **Linux** (kernel version 2.6).
- **Analysis**:
  - This is the **Metasploitable VM**, a purposefully vulnerable machine designed for security testing.
  - **The large number of open ports and outdated services makes it an ideal target for practicing penetration testing techniques**

Why Task 3 Is Important:

- **Understand the Network Landscape**:
  - Task 3 helps create a detailed inventory of devices, their OS, and services.
  - Knowing what devices and software are running allows administrators to manage resources efficiently.
- **Detect Vulnerabilities**:
  - By identifying operating systems and specific service versions, you can pinpoint outdated or misconfigured software with known vulnerabilities.
  - For example, detecting an outdated FTP server (like vsftpd 2.3.4) highlights a potential entry point for attackers.
- **Enhance Security**:
  - Identifying unnecessary or insecure services (e.g., Telnet or outdated Apache HTTPD) allows for proactive remediation, such as updating software or disabling risky services.
- **Compliance and Auditing**:
  - Many regulations (e.g., PCI-DSS, HIPAA) require organizations to document the systems and software running in their networks.
  - Task 3 helps ensure compliance by identifying devices and verifying they are up-to-date and properly configured.
- **Simulate Real-World Threats**:
  - **Attackers perform similar scans to find weak points in a network.**
  - By conducting these scans, network defenders can stay one step ahead, identifying and addressing issues before they are exploited.
- **Troubleshooting and Maintenance**

  Task 3 is useful for identifying misconfigured devices or services that might be causing network issues or performance problems.

This lab demonstrates the process of network discovery and analysis. By identifying devices, operating systems, and services, you gain insight into a network's structure and potential vulnerabilities.