Lab 02
Aim: Network Scanning and vulnerability assessment using NMAP

Vulnerability assessment is a critical component of cybersecurity, helping organizations identify and mitigate potential security weaknesses in their systems and networks. This process typically involves a combination of automated vulnerability scanning tools and manual testing. Here's an overview of both approaches:

1. Vulnerability Scanning Tools:

Automated vulnerability scanning tools are designed to identify known vulnerabilities in software, services, and configurations. These tools scan systems and networks for weaknesses, provide reports, and often offer recommendations for remediation. Some popular vulnerability scanning tools include:

- **Nessus:** Nessus is a widely-used commercial vulnerability scanner known for its extensive vulnerability database and comprehensive scanning capabilities.

- **OpenVAS:** OpenVAS is an open-source vulnerability scanner that provides a range of scanning options and a constantly updated database of known vulnerabilities.

- **Qualys:** Qualys offers cloud-based vulnerability assessment solutions that provide real-time visibility into an organization's security posture.

- **Nexpose**: Nexpose, now known as Rapid7 InsightVM, is another commercial vulnerability scanner with advanced reporting and remediation features.

- **Burp Suite:** Burp Suite is a comprehensive web application security testing tool that can identify vulnerabilities such as SQL injection, cross-site scripting (XSS), and more.

2. Manual Testing:

While automated scanning tools are valuable, they may not detect all vulnerabilities or provide context about the security posture. Manual testing, performed by skilled security professionals, involves in-depth examination of systems and applications. Here are some manual testing techniques:

- **Penetration Testing:** Penetration testers (pen testers) simulate cyberattacks to identify vulnerabilities that automated tools may miss. They use various methods, including ethical hacking, to exploit weaknesses and provide detailed reports.

- **Code Review:** Reviewing the source code of applications and software can reveal vulnerabilities that are not apparent through automated scans. This process involves analyzing the code for potential security flaws.

- **Configuration Review:** Manual review of system configurations, including firewalls, routers, and servers, can uncover misconfigurations that may expose vulnerabilities.

- **Social Engineering Testing:** Assessing the human factor in security by conducting tests like phishing campaigns or physical security assessments to evaluate the effectiveness of security awareness and policies.

- **Manual Web Application Testing:** Skilled testers manually explore web applications, attempting to identify vulnerabilities like SQL injection, XSS, CSRF, and authentication flaws.

- **Wireless Network Testing:** Evaluating the security of wireless networks by attempting unauthorized access and analyzing wireless security configurations.

- **Endpoint Security Assessment:** Manually reviewing endpoint security configurations, analyzing logs, and checking for signs of compromise on individual devices.

3. Combined Approach:

An effective vulnerability management program often combines automated scanning with manual testing. Automated tools can quickly identify common vulnerabilities and provide a baseline assessment, while manual testing offers deeper insights, uncovering complex or unique issues.

4. Continuous Monitoring:

Vulnerability assessment is not a one-time task; it should be an ongoing process. Continuous monitoring, automated scanning, and periodic manual testing are essential to maintaining a strong security posture and addressing new vulnerabilities as they emerge.

Note: Remember that both automated tools and manual testing require skilled professionals who understand security principles, attack techniques, and the specific systems and applications being assessed. Always ensure you have proper authorization before scanning any network or host that you don't own or have permission to scan.

Install Nmap: https://nmap.org/download.html#windows

Open Command Prompt or PowerShell or ZENMAP IDE

1. **Perform a Basic Ping Scan:** To check if the target host is online and responsive, you can use a basic ping scan. This scan sends ICMP echo requests to the target to see if it replies. Open your command prompt or PowerShell and run:

   ```
   nmap -sn 192.168.1.100
   ```

   -sn This option tells Nmap to perform a "ping scan" to determine which hosts are online.

2. **Perform a Full Port Scan:** If you want to scan for open ports on the target, you can use the following command:

   ```
   nmap -p- 192.168.1.100
   ```

   -p- This option tells Nmap to scan all 65,535 ports on the target.

3. **Specify Specific Ports:** If you're interested in scanning specific ports, you can specify them using the -p option. For example, to scan ports 80 (HTTP) and 22 (SSH), you can run:

   ```
   nmap -p 80,22 192.168.1.100
   ```

4. **Save Output to a File:** To save the scan results to a file, you can use the -oN option followed by the filename. For example:

   ```
   nmap -p- 192.168.1.100 -oN scan_results.txt
   ```

   This will save the results in a plain text file named scan_results.txt.

5. **View Results:** After the scan is completed, you can open the scan_results.txt file (or the filename you specified) to view the detailed results of the scan, including open ports and services.

6. **Service Version Detection:** Nmap can attempt to identify the specific versions of services running on open ports. This can be useful for vulnerability assessment and fingerprinting.

   ```
   nmap -sV 192.168.1.100
   ```

   -sV: Enables version detection.

7. **Operating System Detection:** Nmap can attempt to guess the target's operating system based on various network characteristics.

   ```
   nmap -O 192.168.1.100
   ```

   -O: Enables operating system detection.

8. **Aggressive Scan:** The aggressive scan option enables various advanced techniques to find open ports, service versions, and operating system information.

   ```
   nmap -A 192.168.1.100
   ```

   -A: Enables aggressive scanning.

9. **Scan Multiple Targets:** You can scan multiple hosts or IP ranges by specifying them in the command:

   ```
   nmap 192.168.1.100 192.168.1.101
   ```

10. You can also scan a range of IP addresses:

    ```
    nmap 192.168.1.1-50
    ```

11. **Exclude Hosts from Scanning:** Use the -exclude option to exclude specific hosts from the scan:

    ```
    nmap 192.168.1.1-50 --exclude 192.168.1.10
    ```

    This will exclude the host with IP address 192.168.1.10 from the scan.

12. **UDP Port Scan:** By default, Nmap scans TCP ports. To scan UDP ports, use the -sU option:

    ```
    nmap -sU 192.168.1.100
    ```

    This is useful for finding services that use UDP, such as DNS and SNMP.

13. **Timing and Performance Options:** Nmap allows you to control the timing and performance of your scans. You can use options like -T (timing template) and --max-rtt-timeout to adjust scan speed and sensitivity.

    ```
    nmap -T4 --max-rtt-timeout 500ms 192.168.1.100
    ```

14. **Output to XML or Other Formats:** nmap allows you to save scan results in various formats. For example, you can save results in XML format for easier parsing:

    ```
    nmap -oX scan_results.xml 192.168.1.100
    ```

    Other output formats include -oN for normal text, -oG for grepable output, and -oA for all formats.

15. **Scan a Network Range:** You can scan an entire network range by specifying the CIDR notation. For example, to scan all hosts in the 192.168.1.0/24 subnet:

    ```
    nmap 192.168.1.0/24
    ```

16. **Scan for Specific Protocols:** You can specify the protocol you want to scan for using the -sU option for UDP scans and -sT for TCP scans:

    ```
    nmap -sU 192.168.1.100  # UDP scan
    nmap -sT 192.168.1.100  # TCP scan
    ```

17. **Scripting Engine (NSE) Categories:**

Nmap's scripting engine allows you to select specific script categories for more focused scans. These advanced options provide greater flexibility when using nmap for network discovery, vulnerability assessment, and security auditing. Always use nmap responsibly and in compliance with the law and the terms of use of the networks or systems you are scanning. For example, you can scan for "vuln" scripts to check for known vulnerabilities:

```
nmap --script vuln 192.168.1.100
```

NSE Script Categories: You can specify a category to focus on specific aspects of security:

```
  nmap --script discovery <target>
  nmap --script auth <target>
  nmap --script vuln <target>
```

Nmap provides a powerful scripting engine known as the Nmap Scripting Engine (NSE). NSE allows you to write and run custom scripts to perform a wide range of tasks during a network scan. These scripts can be used for various purposes, including vulnerability assessment, service enumeration, and more. NSE scripts are organized into categories based on their functionality. Here are some commonly used NSE script categories and options:

1. Discovery Scripts:

--script discovery: This category includes scripts that help in network discovery, such as finding hosts, services, and information about network devices.

2. Vulnerability Detection Scripts:

--script vuln: These scripts focus on detecting known vulnerabilities in services or systems. They can be used to check for common security issues.

3. Brute Force and Authentication Scripts:

--script auth: Authentication scripts help with authentication bypass and brute-force attacks. They can be used to test the strength of login credentials.

4. Malware and Backdoor Detection Scripts:

--script malware: These scripts can identify known malware and backdoors on a target system.

5. Denial of Service (DoS) Scripts:

--script dos: Scripts in this category can be used to perform various types of Denial of Service attacks or test a system's resilience against such attacks.

6. Exploit and Attack Scripts:

--script exploit: These scripts attempt to exploit vulnerabilities in target services or systems. Be cautious when using these, and ensure you have proper authorization.

7. Web Application Scripts:

--script http-*: NSE includes a variety of scripts for web application scanning, including HTTP enumeration, vulnerability assessment, and more.

8. Database Scripts:

```
  --script mysql-*
  --script oracle-*
  --script sql-*
```

These scripts are used to gather information about database servers and perform various database-related tasks.

9. DNS Scripts:

--script dns-*: These scripts are used for DNS enumeration and testing.

10. SMTP Scripts:

--script smtp-*: These scripts help identify vulnerabilities and issues related to SMTP (email) servers.

11. SMB Scripts:

--script smb-*: NSE scripts for enumerating information from Windows SMB (Server Message Block) services.

12. SSH Scripts:

--script ssh-*: Scripts for SSH enumeration and testing.

13. SNMP Scripts:

--script snmp-*: NSE scripts related to Simple Network Management Protocol (SNMP).

14. SSL Scripts:

--script ssl-*: Scripts to check SSL/TLS configurations and vulnerabilities.

15. Custom Scripts:

You can write your own NSE scripts to perform specific tasks or checks based on your requirements.

Remember to use NSE scripts responsibly and only on systems and networks for which you have proper authorization. Unauthorized or malicious use of NSE scripts can have legal and ethical consequences.

https://www.domaintools.com/

ipconfig /all (to check local host network details)

nslookup (to check IP address or vice versa)

ping