Name- Pratham Vidhani
IBM QRadar AI with Cyber security

Week 2 Assignment

Assignment Overview

In this assignment, we will explore and demonstrate the use of various cybersecurity tools available in Kali Linux. Our goal is to gain practical knowledge and skills in each of the 10 primary categories within Kali Linux, ranging from Information Gathering to Social Engineering Tools. For each category, we will select one tool and delve into its description, use case, and provide step-by-step demonstrations to understand its functionality.

📋 10 Categories Listed in Kali Linux

- Information Gathering
- Vulnerability Analysis
- Web Application Analysis
- Database Assessment
- Password Attacks
- Wireless Attacks
- Reverse Engineering
- Exploitation Tools
- Sniffing Spoofing
- Post Exploitation

All the tools demonstrated and explored in this assignment have been used exclusively on websites on domains that have an active bug bounty program and are intended for bug bounty practice. I want to emphasize that I have the rights and permission to conduct Vulnerability Assessment and Penetration Testing (VAPT) tests on these websites.

Demonstration & Exploration of the tools

#1. Information Gathering (**recon-ng)**

Description:

Recon-ng is a powerful information gathering tool that assists in discovering and collecting data about targets through various reconnaissance techniques.

Use Case:

Recon-ng is particularly useful for the Information Gathering phase, where we aim to gather as much data as possible about our target.

Demonstration Steps:

1. **Setup Recon-ng:**
    - Install Recon-ng on your Linux machine by following the instructions on the official GitHub repository (https://github.com/lanmaster53/recon-ng).
2. **Launch Recon-ng:**
    - Open a terminal and type `recon-ng` to launch the Recon-ng framework.
3. **Configure Modules:-**
    - Recon-ng consists of various modules for different types of information gathering. Configure the modules you plan to use for your specific reconnaissance.
4. **Set the Workspace:**
    - Create a new workspace using the `workspace` command. This helps organize and store the information you gather in a specific project.

    workspace add pratham.vidhani

5. **Add Your Target:**
    - Add your personal website as the target using the `add` command.

    add domains owais-shariff.netlify.app

    ```

6. **Run Modules:**
    - Run the modules you configured in Step 3 to gather information about the target. For example, to perform DNS reconnaissance, you can use the following command:

    use recon/domains-hosts/enumall

    - Set the source to your target (workspace) using the `set` command:

    set SOURCE pratham.vidhani

    - Execute the module using the `run` command:
        run

    - Recon-ng will perform DNS reconnaissance and gather information such as subdomains associated with your website.
7. **Review Results:**
    - Recon-ng will display the results on the screen. You can use commands like `show hosts`, `show domains`, or `show creds` to display specific types of information.
8. **Save Results:**
    - To save the gathered information, use the `db export` command to export the results to a file in your chosen format (e.g., CSV, JSON).

    db export

9. **Explore Other Modules:**
    - Depending on your specific information gathering needs, explore and run other Recon-ng modules for web crawling, scraping, or other reconnaissance tasks.
10. **Exit Recon-ng:**
    - When you're done, exit Recon-ng using the `exit` command
    ```

Example run using hackertarget as the module

2. Vulnerability Analysis (**nmap)**

Description:

Nmap (Network Mapper) is a versatile open-source tool used for network discovery and security auditing. It can discover hosts and services on a network and identify potential vulnerabilities.

Use Case:

Nmap is essential for Vulnerability Analysis as it helps in identifying open ports, services running on those ports, and potential weaknesses.

Demonstration Steps:

1. **Install Nmap:**
    - Begin by installing Nmap on your computer. You can download it from the official website (https://nmap.org/download.html) or use a package manager for your operating system.
2. **Determine the Target:**
    - Identify the target or network that you want to perform vulnerability analysis on. This could be a single host, a range of IP addresses, or an entire network.
3. **Scan for Open Ports:**
    - Use Nmap to scan for open ports on the target. This information will help you understand which services are running and potentially vulnerable.

```
nmap -p- -T4 <target>
```

4. **Identify Service Versions:**
    - After finding open ports, use Nmap to identify the versions of services running on those ports. Knowing the service versions can help in searching for known vulnerabilities associated with specific versions.

```
nmap -sV <target>
```

5. **Perform a Vulnerability Scan:**
    - Use Nmap in combination with a vulnerability scanning script or database to identify potential vulnerabilities on the target system. Nmap scripts (NSE scripts) can be used for this purpose.

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

6. **Output Results:**
    - Save the results of your Nmap scans to a file for further analysis. You can use the `oN` flag to specify the output format and file name.

```
nmap -oN scan_results.txt <target>
```

7. **Analyze Results:**
    - Examine the results of your Nmap scans to identify potential vulnerabilities and security issues. Pay attention to open ports, service versions, and any specific vulnerabilities reported.

Example Nmap Scan of My Website that reveals that port 80 TCP is open

3. Web Application Analysis (**wpscan)**

Description:

WPScan is a specialized tool for WordPress vulnerability scanning and enumeration, making it ideal for analyzing web applications built on the WordPress platform.

Use Case:

WPScan is valuable in Web Application Analysis, particularly when dealing with WordPress-based websites.

Demonstration Steps:

1. **Install WPScan:**
    - Begin by installing WPScan on your system. WPScan is a Ruby-based tool, so ensure you have Ruby installed first. Then, install WPScan using the following commands:

    ```
    gem install wpscan
    ```

2. **Run a Basic Scan:**
    - To perform a basic scan of a WordPress website, use the following command:

    ```
    wpscan --url <target-website>
    ```

    Replace `<target-website>` with the URL of the WordPress website you want to scan. This command will initiate a scan with default settings.

3. **Customize the Scan:**
    - You can customize the scan by adding various options and flags to the `wpscan` command. For example, you can specify a specific plugin or theme to target, increase the verbosity of the output, or enable debugging.
4. **Output Results:**
    - WPScan will display the scan results on the screen, including information about the WordPress version, installed themes, plugins, vulnerabilities, and more.

`[elementor.com](http://elementor.com)` is an example of a site that uses wordpress, you can see the technologies that a website uses by utilizing tools like wappalyser.

Critical information like out of date version, plugins used etc has been revealed.

4. Database Assessment (**sqlmap)**

Description:

SQLMap is a powerful tool for automated SQL injection and database takeover attacks. It's essential for assessing the security of databases.

Use Case:

SQLMap is indispensable in Database Assessment to uncover SQL injection vulnerabilities.

Demonstration Steps:

1. **Install SQLMap:**
    - Start by installing SQLMap on your system. You can download the latest version from the official GitHub repository (https://github.com/sqlmapproject/sqlmap) or use a package manager if available for your operating system.
2. **Identify the Target:**
    - Determine the target website or application that you suspect might be vulnerable to SQL injection.
3. **Perform a Basic Scan:**
    - Use SQLMap to perform a basic scan against the target URL to detect potential SQL injection vulnerabilities. The basic syntax is:

```
sqlmap -u <target-URL>
```

    Replace `<target-URL>` with the URL of the target web page or application.

4. **Specify Other Scan Parameters:**
    - SQLMap provides numerous options to customize the scan, including the ability to specify parameters, payloads, and various detection techniques. You can use options like `p` to specify parameters, `D` to target a specific database, and `T` to target a specific table.
5. **Select the Injection Point:**
    - SQLMap will analyze the target and attempt to identify potential injection points. You can select the injection point interactively or specify it using the `-data` option for POST requests or `-url` for GET requests.
6. **Run the Full Scan:**
    - Once you've selected the injection point and customized the scan parameters, run the full scan by using the `-dbs` option to enumerate databases:

```
sqlmap -u <target-URL> --dbs
```

7. **Enumerate Databases and Tables:**
    - SQLMap will attempt to enumerate databases, tables, and other information based on the detected injection point. You can use options like `-dbs` to list databases, `-tables` to list tables in a specific database, and so on.

8. **Dump Data:**
    - To extract data from a specific table, use the `-dump` option followed by the database and table names. For example:

    sqlmap -u <target-URL> -D <database-name> -T <table-name> --dump

9. **Exploit Other Options:**
    - SQLMap provides advanced options for exploiting SQL injection vulnerabilities, such as privilege escalation and shell access. Be cautious when using these options and ensure you have proper authorization to exploit vulnerabilities.
10. **Output Results:**
    - SQLMap will display the results of the scan on the screen, including information about the detected vulnerabilities, database names, table names, and the extracted data.
11. **Save Results:**
    - If desired, you can save the scan results to a file using the `o` option:

    sqlmap -u <target-URL> --dbs -o results.txt

Here is the query I used to get the results above, I added a few extra flags to get a dump of the names of artists on a pentesting site.

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T artists -C aname --dump
```

### 5. Password Attacks (**johntheripper)**

**Description:**

John the Ripper is a widely-used password cracking tool that helps assess password security by attempting to crack password hashes.

**Use Case:**

John the Ripper is employed for Password Attacks to test the strength of passwords in various formats.

**Demonstration Steps:**

1. **Install John the Ripper:**
    - Begin by installing John the Ripper on your system. You can download the latest version from the official website ([https://www.openwall.com/john/](https://www.openwall.com/john/)) or use a package manager if available for your operating system.
2. **Download Wordlists (Optional):**
    - To perform password cracking, you'll often need a wordlist containing potential passwords. You can obtain wordlists from various sources, or you can create custom wordlists based on your needs.
3. **Create a Password Hash File:**
    - To crack passwords, you need a file containing password hashes. This file can be obtained from various sources, such as a system's shadow file, database dumps, or other sources.
4. **Run John the Ripper:**
```

    - To start using John the Ripper, open a terminal and navigate to the directory where the program is installed or where the binary is located. Use the following command to specify the hash file and the wordlist:

    john --wordlist=<wordlist-file> <hash-file>

    Replace `<wordlist-file>` with the path to your wordlist and `<hash-file>` with the path to your password hash file.

5. **Monitor Progress:**
    - John the Ripper will start attempting to crack the passwords in the hash file using the provided wordlist. It will display progress information, including cracked passwords, on the terminal.
6. **View Cracked Passwords:**
    - Once John the Ripper successfully cracks any passwords, it will display them on the screen. You can find the cracked passwords in the program's output.
7. **Customize Cracking Rules (Optional):**
    - John the Ripper provides various cracking rules and modes that you can customize to improve your cracking results. You can experiment with these rules to optimize your cracking process.
8. **Save Results (Optional):**
    - If you want to save the results to a file for further analysis or documentation, you can use the `-show` option to display cracked passwords and redirect the output to a file:

    john --show <hash-file> > results.txt

9. **Use Community and Community-Powered Wordlists (Optional):**
    - In addition to your own wordlists, you can leverage community-contributed wordlists and cracking rules to increase your chances of success.
10. **Cleanup and Secure Hashes (Important):**
    - It's essential to handle password hashes with care and ensure that you have the necessary permissions and authorization to perform password cracking. After you're done, make sure to securely store and manage the obtained password information.

I then put john the ripper to the test by running this file thru it and adding an output operator pointing to a file named `hash.txt`

On opening it with vim, `vim hash.txt`, I confirmed that john the ripper had cracked the successful value and put it in the hash.txt file in the form of a hash.

6. Wireless Attacks (**aircrack-ng)**

Description:

Aircrack-ng is a comprehensive suite of tools used for wireless network analysis and security assessment.

Use Case:

Aircrack-ng is ideal for Wireless Attacks, especially in scenarios where you want to assess the security of Wi-Fi networks.

Demonstration Steps:

1. Start Aircrack-ng tools, such as airodump-ng, aireplay-ng, and aircrack-ng.
2. Capture data packets from a target Wi-Fi network.
3. Launch a dictionary attack or perform deauthentication attacks to crack the Wi-Fi password.
4. Evaluate the success of the attack and the network's security.

Please note that most of these require a baremetal/dualboot install of linux with a physical device that supports monitor mode. Due to the inaccessiblity of such a device, I haven’t demonstrated this project.

7. Reverse Engineering (**radare2)**

Description:

Radare2 is a powerful open-source framework for reverse engineering and binary analysis.

Use Case:

Radare2 is valuable in Reverse Engineering for dissecting and understanding the functionality of binaries and malware.

Demonstration Steps:

1. Launch Radare2 and load a binary for analysis.
2. Explore and disassemble the binary code.
3. Investigate functions, control flow, and potential vulnerabilities.
4. Document findings and insights from the analysis.

I created a binary called crackme by running a C language code using `gcc test.c -o crackme` then opened it in r2 to read binaries.

One can find the entry point of the binary using `ie` and main address using `iM` as below:

Also, the binary is not stripped.

Analyzing this information is important as binaries can have a lot of subtleties that can prevent one from debugging the same. Knowing the same will help us come up with workarounds around the protections.

8. Exploitation Tools (m**etasploit framework)**

Description:

The Metasploit Framework is a widely-used penetration testing tool that helps identify, exploit, and validate vulnerabilities in target systems.

Use Case:

Metasploit Framework is crucial in Exploitation Tools to simulate real-world attacks and test system defenses.

Demonstration Steps:

1. Open the Metasploit console in Kali Linux.
2. Select a module for a specific vulnerability.
3. Configure the exploit parameters and set the target.
4. Execute the exploit and gain access to the target system.
5. Validate the vulnerability and assess the impact.

Example Usage: Gaining Root Access on a Localhost with a payload listed

9. Sniffing Spoofing (Wireshark**)**

Description:

Wireshark is a widely-used network packet analyzer that allows you to capture and inspect network traffic in real-time.

Use Case:

Wireshark is essential in Sniffing Spoofing to intercept and analyze network packets, making it valuable for network troubleshooting and security analysis.

Demonstration Steps:

1. Launch Wireshark on your system.
2. Select a network interface to capture traffic (e.g., Ethernet, Wi-Fi).
3. Start capturing packets to analyze network communication.
4. Apply filters to focus on specific protocols or traffic patterns.
5. Examine captured packets to identify potential anomalies or security threats.

Here’s an example test I performed on a wireshark capture file that I found online. By following the TCP Trace, I was able to piece together the HTML rendering and see what exactly the user saw. This happened because the connection was over HTTP instead of the much more secure HTTPS.

10. Post Exploitation (**mimikatz)**

Description:

Mimikatz is a powerful post-exploitation tool used to extract and manipulate credentials from compromised Windows systems.

Use Case:

Mimikatz is critical in Post Exploitation scenarios to escalate privileges and gather sensitive information.

Demonstration Steps:

1. **Download and Extract Mimikatz:**
    - Download the latest Mimikatz release from the official GitHub repository

(https://github.com/gentilkiwi/mimikatz).
    - Extract the contents to a directory on your Windows system, e.g., "C:\Mimikatz."
2. **Open an Elevated Command Prompt:**
    - On your Windows system, press `Win + X` and choose "Windows Terminal (Admin)" or "Command Prompt (Admin)" to open an elevated command prompt.
3. **Navigate to the Mimikatz Directory:**
    - Use the `cd` command to navigate to the directory where you extracted Mimikatz. For example:

```
cd C:\\Mimikatz
```

4. **Load the Mimikatz Module:**
    - To use Mimikatz, you need to load its module. Run the following command:

```
mimikatz.exe
```

5. **Dump Credentials from Memory:**
    - Inside the Mimikatz shell, you can use various commands to extract credentials. For example, to extract plaintext passwords from memory, use:

```
sekurlsa::logonpasswords
```

6. **View Extracted Credentials:**
    - Mimikatz will display the extracted credentials, including usernames and plaintext passwords, if available.
7. **Additional Commands:**
    - Explore other Mimikatz commands to perform various post-exploitation activities, such as listing loaded DLLs, Kerberos ticket extraction, and more.

Note: This requires a vulnerable windows system. Mimikatz was built for Windows 7 and even worked till Windows 10 on some version. However, As of today for my system, Windows 11, This security vulnerability has been patched and is no longer effective. Which is why I will note be demonstrating this myself.