**Lab 9-01: Perform web application Reconnaissance using Nmap and Telnet**

**Scenario**

You work in a software company and have to audit its numerous clients’ web applications. You have to gather valuable information about their web applications to be used for additional testing. Web infrastructure footprinting helps you to identify vulnerable web applications, understand how they connect with peers and the technologies they use, and find vulnerabilities in specific parts of the web app architecture. These vulnerabilities can further help you to exploit and gain unauthorized access to web applications.

**Solution**

You will use nmap and telnet tools to gather helpful information about web applications. In web application reconnaissance, you perform various tasks such as server discovery, service discovery, server identification or banner grabbing, and hidden content discovery. A professional, ethical hacker or pen tester must gather as much information as possible about the target website by performing web application footprinting using various techniques and tools.

**Note:** Before port scanning, perform a Whois lookup to gather information about the IP address of the web server and the complete information about the domain, such as its registration details, name servers, IP address, and location. Use tools such as Netcraft (https://www.netcraft.com), SmartWhois (https://www.tamos.com), WHOIS Lookup (https://whois.domaintools.com), and Batch IP Converter (http://www.sabsoft.com) to perform the Whois lookup. Perform DNS Interrogation to gather information about the DNS servers, DNS records, and types of servers used by the target organization. DNS zone data includes DNS domain names, computer names, IP addresses, domain mail servers, service records, etc. Use tools such as DNSRecon (https://github.com) and Domain Dossier (https://centralops.net) to perform DNS interrogation.

|  |
| --- |
| 1. Turn on the Parrot Security virtual machine. Open a **Terminal** window and execute **sudo su** to run the programs as a root user. Then, run the **cd** command to jump to the root directory. To perform a port and service discovery scan, run the **nmap -T4 -A -v [Target Web Application]** command (here, the target web application is [**www.certifiedhacker.com**](http://www.certifiedhacker.com)).  **-T4**: specifies setting time template (0-5),  **-A:** specifies aggressive scan,  **-v:** enables the verbose output (includes all hosts and ports in the output)    2. The result appears, displaying the open ports and services running on the machine hosting the target website.    3. To identify the make, model, and version of the target web server software, we will perform banner grabbing. Run the command **telnet www.certifiedhacker.com 80** to establish a telnet connection with the target machine.  **Note:** Port 80 is the port number assigned to the commonly used internet communication protocol, Hypertext Transfer Protocol (HTTP).The **Trying 162.241.216.11**... message appears; type **GET / HTTP/1.0** and press **Enter** two times.    4. The result appears, displaying information related to the server name and its version of technology used.    5. This concludes the demonstration of how to perform web application reconnaissance. |

## Lab 9-02: Perform Web Spidering using OWASP ZAP

**Scenario**

You work in a software development company, which is concerned about potential security flaws in its newly developed web application before deploying it to production. As part of the security assessment, you have been tasked with mapping the application structure using the Open Web Application Security ProjectZed Attack Proxy (OWASP ZAP). Web spidering will be used to discover all accessible endpoints, pages, directories, and parameters within the application. This information will help you identify attack surfaces and potential vulnerabilities in the application.

**Solution**

OWASP ZAP is a comprehensive penetration testing tool designed to identify vulnerabilities in web applications. It includes automated scanners and a variety of tools that enable users to detect security issues manually. ZAP caters to different skill levels, making it suitable for developers, testers who are new to security testing, and specialists in security testing.

|  |
| --- |
| 1. Turn on the Parrot Security machine, open a **Terminal** window, and execute **sudo su** to run the programs as a root user. Then, run the **cd** command to jump to the root directory.  2. In the **Terminal** window, type **zaproxy** and press **Enter** to launch OWASP ZAP.    3. The OWASP ZAP initializing window opens; wait for it to complete. After initialization, a prompt that reads **Do you want to persist the ZAP Session?** appears; select the **No, I do not want to persist this session at this moment** **in time** radio button and click **Start**.    4. The main window of OWASP ZAP opens. Under the **Quick Start** tab in the Welcome to OWASP ZAP section, select the **Automated Scan** option.    5. In the **Automated Scan** wizard, enter the target website under the **URL to attack** field (here, **www.testphp.vulnweb.com/.com**). Leave the other settings to default and click the **Attack**.    6. OWASP ZAP starts scanning the target website. You can observe various URLs under the **Spider** tab. After performing web spidering, OWASP ZAP performs **active scanning**. Navigate to the **Active Scan** tab to observe the various scanned links.    7. After completing the active scan, the results appear under the **Alerts** tab, displaying the various vulnerabilities and issues associated with the target website. The objective is web spidering, so we will focus on the information obtained while performing web spidering.    8. Now, click on the **Spider** tab from the lower section of the window to view the web spidering information. By default, the URLs tab appears under the **Spider** tab. The URLs tab contains various links for hidden content and functionality associated with the target website ([**www.testphp.vulnerweb.com**](http://www.testphp.vulnerweb.com)).  s  9. Now, navigate to the **Messages** tab under the **Spider** tab to view more detailed information regarding the URLs obtained while performing the web spidering, as shown in the screenshot.  **Note:** In real-time, attackers perform web spidering or crawling to discover hidden content and functionality that is not reachable from the main visible content to exploit user privileges within the application. It also allows attackers to recover backup copies of live files, configuration, and log files containing sensitive data, backup archives containing snapshots of files within the web root, and new functionality that is not linked to the main application.    14. Close all open windows and document all acquired information. This concludes the demonstration of how to perform web spidering on a target website using OWASP ZAP. |

**Lab 9-03: Perform Web Application Vulnerability Scanning using SmartScanner**

**Scenario**

You work in a software company and are tasked to assess the security of a web application owned by a financial institution. The organization wants to ensure that its web application is free from vulnerabilities that could lead to unauthorized data breaches or attacks. You will use **SmartScanner**, a powerful vulnerability scanning tool, to identify potential security issues.

**Solution**

You will use SmartScanner, which utilizes Machine Learning (ML) and Artificial Intelligence (AI) techniques to tailor its approaches based on the target's behavior. This integration helps minimize false positives. SmartScanner employs AI to identify vulnerable pages, detect custom 404 pages, recognize input vectors, fingerprint the target, and assess security risks. In this context, we will explore vulnerabilities in the target web application using SmartScanner.

|  |
| --- |
| 1. Open the Windows machine and log in using your credentials. Open a web browser and download **thesmartscanner** from the given link [**https://www.thesmartscanner.com/downloads**](https://www.thesmartscanner.com/downloads).    2. After downloading, click the **smart scanner** to launch it.  **Note:** If you get a pop-up from **Microsoft Defender**, click on **Run anyway**.    3. In the license agreement, click on **I accept the agreement** and click on **Next**.    4. In ready to install window, click on **Install**.    5. In **Do You want** to allow this app to make changes to your device, click on **Yes**.  6. Click on **Finish.** The smartscannner is successfully downloaded.    7. Now open SmartScanner. In the **enter site address to scan** field, enter **[TARGET WEBSITE]** and click the **Scan** button.    9. The tool starts scanning the target website for vulnerabilities.    10. Once the tool completes scanning, it will display the issues found under the **Found Issues** section and the **Severity of the Issues**.    11. Now, expand Password Sent Over HTTP and click on the first **http://testphp.vulnweb.com/login.php** link from the left pane to view the details of the vulnerability.    12. In the right pane, scroll down to the **DESCRIPTION** section. This website contains a vulnerability that attackers could exploit to intercept sensitive information like passwords during transmission over unencrypted HTTP traffic. You can also go through the **RECOMMENDATION** section to check for the recommended actions to patch the vulnerability.    13. Now, under the **CLASSIFICATION** section, press **Ctrl** and click on the **CWE-319** hyperlink. A **CWE** website appears in the **Microsoft Edge** web browser, displaying the details of **CWE-319 ClearText Transmission of Sensitive Information**.    14. Similarly, click the **http://testphp.vulnweb.com** link available under the **X-Frame-Options Header is Missing** node, which is termed as **Low** severity.    15. Scroll down to the **DESCRIPTION** here. We can observe that the X-Frame-Options Header is missing, which will make this site vulnerable to click-jacking.    16. Similarly, you can view the **RECOMMENDATION** section and click on the reference link under the REFERENCES section for further analysis.    17. Expand the **X-Content-Type-Options Header is Missing node** and click on the **http://testphp.vulnweb.com** link to view its contents. Under the **DESCRIPTION** section, we can observe that browsers can perform **MIME sniffing**, which can cause them to transform non-executable content into executable content. Similarly, you can view the **RECOMMENDATION** section and click on the reference link under the **REFERENCES** section. You can also click on any other vulnerability to view its detailed information.    18. You can also save the report for further analysis. This concludes the demonstration of discovering vulnerabilities in a target website using SmartScanner. |

**Lab 9-04: Perform a Brute-force Attack using Burp Suite**

**Scenario**

You work in a software company as an ethical hacker. The company suspects that its web application may have weak authentication mechanisms, making it vulnerable to brute-force attacks. As an ethical hacker, you have been tasked to test the login functionality of the web application using Burp Suite to identify potential vulnerabilities in its authentication system. The objective is to simulate an attack where a brute-force technique is used to guess valid login credentials by systematically testing multiple username and password combinations.

**Solution**

You will use Burp Suite, which is a comprehensive platform designed for the security testing of web applications. It includes a variety of tools that collaborate to facilitate the entire testing process, from the initial mapping and analysis of an application’s attack surface to identifying and exploiting security vulnerabilities. Key components of Burp Suite include an intercepting proxy, an application-aware spider, an advanced web application scanner, an intruder tool, a repeater tool, and a sequencer tool.

|  |
| --- |
| 1. Log in to the Parrot OS machine and set up a **Burp Suite** proxy by first configuring the proxy settings of the browser. In the Mozilla Firefox browser, click the **Open application menu** icon in the right corner of the menu bar and select **Settings** from the drop-down list.    2. The **General settings** tab appears. In the **Find in Settings** search bar, search for **proxy,** and in the **Search Results**, click the **Settings** button under the **Network Settings** option.    3. In the Connection Settings window, select the **Manual proxy configuration** button and specify the **HTTP Proxy** as **127.0.0.1** and the Port as **8080**. Tick the **Also use this proxy for HTTPS** checkbox and click **OK**. Close the **Settings** tab and minimize the browser window.    4. To launch the Burpsuite CE application, click the **Applications** menu from the top left corner of the **Desktop** and navigate to **Pentesting** → **Web Application Analysis** → **Web Application Proxies** → **Burpsuite CE**.    5. The Burp Suite Community Edition pop-up appears; click **OK**. In the **Terms and Conditions** wizard, click the **I Accept** button. The Burp Suite main window opens; ensure that the **Temporary project** button is selected and click on **Next.**    6. In the next window, select the **Use Burp defaults** radio button and click the **Start Burp** button.    7. In the Burp Suite main window, click the **Proxy** tab from the available options in the top section of the window.    8. In the **Proxy** settings, by default, the **Intercept** tab opens up. Observe that by default, the interception is active as the button says **Intercept is on**. Leave it running.**Note:** Turn the interception on if it is off.    9. Switch back to the browser window. On the login page of the target WordPress website, type random credentials and click the **LOG IN** button.    10. Switch back to the **Burp Suite** window; observe that the application interceptedthe **HTTP** request. Now, right-click anywhere on the HTTP request window, and from the context menu, click **Send to Intruder**. Observe that Burp Suite intercepted the entered login credentials.    11. Now, click on the **Intruder** tab from the toolbar and observe that under the **Intruder** tab, the **Positions** tab appears by default. In the **Positions** tab under the Intruder tab, observe that Burp Suite sets the target positions by default, as shown in the HTTP request. Click the **Clear §** button in the right pane to clear the default payload values.    12. Once you clear the default payload values, select **Cluster bomb** from the **Attack type** drop-down list.    13. Now, we will set the username and password as the payload values. To do so, select the **username** value you entered and click **Add §** from the right pane. Similarly, select the **password** value you entered and click **Add §** from the right pane.    14. Once the username and password payloads are added. The symbol **‘§’** will be added at the start and end of the selected payload values.    15. Navigate to the **Payloads** tab under the **Intruder** tab and ensure that under the **Payload Sets** section, the **Payload set** is selected as **1**, and the **Payload type** is selected as **Simple list**. Under the **Payload settings [Simple list]** section, click on **Load...**. A file selection window appears; navigate to the location **/home/parrot/Desktop/username,** select the **username.txt** file, and click the **Open** button.    16. The selected **username.txt** file content appears under the **Payload settings [Simple list]** section.    17. Similarly, load a password file for the payload set **2**. From the **Payload Sets** section, select the **Payload set** as **2** from the drop-down options and ensure that the **Payload type** is selected as **Simple list**. Under the **Payload settings [Simple list]** section, click the **Load...** button. A file selection window appears; navigate to the location **/home/parrot/Desktop/password**, select the **password.txt** file, and click **Open.**    18. Observe that the selected **password.txt** file content appears under the **Payload settings [Simple list]** section.    19. Once the wordlist files are selected as payload values, click the **Start attack** button to launch the attack.    20. A **Burp Intruder** notification appears. Click **OK** to proceed.  21. The **Intruder attack** window appears as the brute-attack initializes. It displays various username-password combinations along with the **Length** of the response and the **Status**. Wait for the progress bar at the bottom of the window to complete. After the progress bar completes, scroll down and observe the different values of **Status** and **Length**. Here, Status=400 and Length= 726.**Note:** Different values of Status and Length indicate that the combination of the respective credentials is successful.    22. In the **Raw** tab under the **Request** tab, the HTTP request with a set of the correct credentials is displayed. (here, username=admin123 and password=qwerty). Note down these user credentials.  **Note:** If the actual username and password combination are not present in your username and password file. Your credentials for login will not work.    23. Navigate back to the **Proxy** tab and click the **Intercept is on** the button to turn off the interception. The **Intercept is on** button toggles to **Intercept is off.** Switch to the browser window and remove the browser proxy set up by selecting **No proxy** in the **Connection Settings window,** and click **OK**. Close the tab.24. Reload the target website, enter the **Username** and **Password** obtained in Step#22, and click **Log In**. You will be able to log in using the brute-forced credentials successfully.    25. This concludes the demonstration of how to perform a brute-force attack using Burp Suite. |

**Lab 9-05: Perform Remote Code Execution (RCE) Attack**

**Scenario**

You work in a software company as a penetration tester. The company suspects that its WordPress application is vulnerable due to a known flaw in the wp-upg plugin, which can potentially allow attackers to perform Remote Code Execution (RCE) attacks. As a penetration tester, you have been assigned to exploit this vulnerability in a controlled environment to verify its presence and assess the risk level. The objective is to simulate an RCE attack and analyze the security implications for the organization's web application.

### Solution

You will exploit the RCE vulnerability in the wp-upg plugin by injecting and executing malicious code on the target system, ultimately identifying how the attacker could gain unauthorized access and control over the web application.

|  |
| --- |
| 1. To download the wampserver, go to the website: [**https://sourceforge.net/projects/wampserver/**](https://sourceforge.net/projects/wampserver/). Click on the **Download** button. The download will start automatically within a few seconds. SourceForge may redirect you to a list of available WampServer versions. Select the latest version compatible with your system (32-bit or 64-bit).    2. Go to the **Downloads** folder and double-click the **wampserver.exe** file to start the installation process. Select **English** as the language and click on **OK**.    3. In **Setup**, select **I ACCEPT THE AGREEMENT** and click on **Next**.    4. Click on **Next.**    5. A dialog-box appears that tells you to download certain packages to proceed.    6. Go to the link [**https://wampserver.aviatechno.net/**](https://wampserver.aviatechno.net/) scroll down to **Visual C++ Redistribution Packages,** anddownload the packages that are required.    7. After that, go to **Download**s and click on the downloaded file to install the **Microsoft Visual C++ Redistribution.**    8. Now, go back to the Wamp server, specify the location in **Select Destination Location**, and click on **Next**.    9. Click on **Next.**    10. Click on **Finish**. Hence, the wampserver is successfully installed.    11. Now, launch the Wampserver. In the right corner of the **Desktop**, click the **Show hidden icons** icon and observe that the WampServer icon appears. Wait for this icon to turn **green**, which indicates that the **WampServer** is successfully running.    12. Now, open any web browser and go to **http://10.10.1.22:8080/CEH/wp-login.php?** A WordPress webpage appears. Type your **Username** or **Email Address** and **Password,** and click on the **Log In** button.  **Note**: Here, you can use any wordpress based website. We are using the above-mentioned website as the victim.    13. Assume that you have installed and configured the User Post Gallery plugin.  **Note:** you can also see the following links: [**https://wordpress.org/plugins/easy-post-gallery/**](https://wordpress.org/plugins/easy-post-gallery/)  **https://codecanyon.net/item/user-gallery-wordpress-plugin/16122370?srsltid=AfmBOop7FhkLx4cBLZJMk-oe16cJeMhZEUY3cCaCQv4W8lb0vjOb3ywi**  14. Hover your mouse cursor on **Plugins** in the left pane and click **Installed Plugins.**    15. In the **Plugins** page, observe that the **User Post Gallery** is installed. Click **Activate** under the **User Post Gallery** plugin to activate the plugin.    16. Hence, the plugin is activated.    17. Now, Switch to the Parrot Security machine. Open a web browser, go to **https://wpscan.com/**, and log in to the wpscan account. In the **Profile,** observe the **API Token**. Note down or copy this API Token; we will use this token in the later steps.  **Note:** If you do not have an account, sign up for the wpscan website to create one.    18. Now, open a **Terminal** window and execute **sudo su** to run the programs as a root user. Now, run the **cd** command to jump to the root directory. Run the command **wpscan --url http://10.10.1.22:8080/CEH --api-token [API Token from Step#17].**    19. The result appears, displaying detailed information regarding the target website.    20. Scroll down to the **Plugin(s) Identified** section and observe the installed vulnerable plugins (wp-upg) on the target website. In the **Plugin(s) Identified** section, within the context of the wp-upg plugin, an Unauthenticated Remote Code Execution (RCE) vulnerability has been detected, as shown in the screenshot.**Note:** The number of vulnerable plugins might differ when you perform this lab.    21. We will exploit the RCE vulnerability present in the wp-upg plugin. To perform an RCE attack, run the command **curl -i 'http://10.10.1.22:8080/CEH/wp-admin/admin-ajax.php?action=upg\_datatable&field=field:exec:whoami:NULL:NULL'.** This curl command exploits a WordPress plugin vulnerability by sending a malicious request to the **admin-ajax.php** file, allowing an attacker to execute arbitrary system commands via the exec function, potentially leading to remote code execution. In the last step, the **whoami** command was executed, yielding the outcome **nt authority\ \system.**    22. This concludes the demonstration of performing an RCE attack. |

**Lab 9-06: Detect Web Application Vulnerabilities using Wapiti Web Application Security Scanner**

**Scenario**

You work in a financial company as a security manager. The company has launched a customer-facing web application to handle online transactions. The company is concerned about potential vulnerabilities like SQL injection, XSS, and insecure configurations that could compromise customer data or application security. To address these concerns, the company asked you to identify and mitigate vulnerabilities in the web application. You will use the Wapiti web application vulnerability scanner that performs black-box testing by crawling the website and scanning for known security weaknesses.

**Solution**

You will use Wapiti to conduct a thorough vulnerability scan. First, the tool is configured by providing the target URL of the web application. Wapiti crawls the website to identify all endpoints, forms, and input fields. It then simulates attacks like SQL injection and XSS to detect vulnerabilities. Once the scan is complete, Wapiti generates a detailed report highlighting security issues, including insecure configurations and injection points. The security team reviews the report, prioritizes critical vulnerabilities, and remediates them by implementing secure coding practices, proper input validation, and sanitization. A Web Application Firewall (WAF) is also deployed to block malicious traffic, and regular security audits are scheduled to maintain ongoing protection.

|  |
| --- |
| 1. Turn on the Parrot Security virtual machine and log in with your credentials. Open a Terminal window and install wapiti by using the commands **sudo apt update** and **sudo apt install wapiti.**      2. Check the installation path by typing **which wapiti.** Enter **sudo su** to enter the root directory, then type **pip show wapiti3.** This will show allinstallation information. You can also confirm installation by typing the command **wapiti --version** into the terminal.    3. After installing the tool, run the **wapiti -u https://www.certifiedhacker.com** command to perform web application security scanning on the certifiedhacker.com website.      4. Now, in the terminal, run **cd /root/.wapiti/generated\_report/** to navigate to the generated\_report directory. Run **the ls** command to view the contents of the directory. We can see that the **certifiedhacker.com\_12212024\_1511.html** file has been created.    5. Run **cp certifiedhacker.com\_12212024\_1511.html /home/parrot/** command to copy the HTML file to **/home/parrot** location    6. Open a new terminal and run the **firefox certifiedhacker.com\_12212024\_1511html** command to open the HTML file in the Firefox browser.    7. The Wapiti scan report opens up in the Firefox browser. You can analyze the scan result with the discovered vulnerabilities. Scroll down to view the detailed information regarding each discovered vulnerability.    8. This concludes the demonstration of discovering vulnerabilities in a target website scanning using Wapiti. |

**Lab 9-07: Perform an SQL Injection Attack Against MSSQL to Extract Databases using sqlmap**

**Scenario**

You are hired as an ethical hacker to test some database vulnerabilities. You are required to simulate SQL injection attacks to retrieve hidden data. You can use manual methods to find web application vulnerabilities in SQL injection (SQLi).

**Solution**

You must use some techniques to find SQLi vulnerabilities in online applications. You will need to inject some SQL queries with the help of the commenting functionality in SQL. You will use different payloads for this; hence, if one payload does not work, you can try another.

|  |
| --- |
| 1. The following full URL is used as an example for the lab: [**http://testphp.vulnweb.com/listproducts.php?category=2**](http://testphp.vulnweb.com/listproducts.php?category=2)    2. Right-click on the page and select **Inspect (Q)**.    3. A new window appears at the bottom. Select **Console**. On the console tab, write **document.cookie** and press enter. It will show the cookie value; copy this value.    4. On ParrotOS, open the terminal and write the command **sudo su** to operate in root mode.  5. Use the command pattern **sqlmap -u "URL" --cookie="value" –dbs to launch sqlmap attack.** Copy the URL from **Step#1** and the cookie value from **Step#3** and paste it here.    In this step, **-u** specifies the target URL. The URL contains a parameter **cat=2** that is being tested for SQL injection. The **--cookie** option sends a custom HTTP cookie to simulate an authenticated session. Here, the cookie value is **login=test%2Ftest**, which may bypass authentication checks. The **--dbs** option tells SQLMap to enumerate and list all available **databases** on the target server.    6. If it asks to keep testing, enter **y** and press **Enter.**    7. Hence, it provides the databases and their information.    8. Use the command **sqlmap -u <URL> -D db\_name --tables** command to scan the tables in the database.    9. It shows how many tables are present in the database and their names.    10. Use the command **sqlmap -u <URL> --cookie=<value> -D <db name> -T**  **<table name> --dump** command to scan the tables in the database.    11. The result shows that there is one entry in this particular table that shows all the details, such as name, password, email, phone number, etc.    12. You can also use the command **sqlmap -u <URL> --cookie=<value> --os-shell** to retrieve the hostnames of the machine on which the target web application is running.    13. This concludes the end of sqlmap. |