MODULE 14 Hacking Web Applications
LAB REPORT

Lara Alofi

VULNERABILITY ANALYSIS AND DEFENSE
Course Code: T44-17520

Lab Session Identifiers

https://eccouncil.learnondemand.net/Lab/Launch/55236?AssignmentId=1340522&lang=

https://labclient.labondemand.com/LabClient/036252ac-81bc-47db-81f0-1ba13905006a

https://labclient.labondemand.com/LabClient/5da41e02-66e6-44da-865b-63ea55b96f92

Username on EC-Council System

1. 2110886@uj.edu.sa

Lab 01: Footprint the Web Infrastructure

Task 1: Perform Web Application Reconnaissance using Nmap and Telnet

Steps:

1. Log into Parrot Security Machine:
    - Click the Parrot Security machine.
    - Enter the username and password (toor) to log in.
2. Whois Lookup:
    - Use tools like Netcraft, SmartWhois, WHOIS Lookup, and Batch IP Converter to gather domain registration details, name servers, IP addresses, and location.
3. DNS Interrogation:
    - Use DNSRecon, DNS Records, and Domain Dossier to collect information about DNS servers, DNS records, and types of servers.
4. Port Scanning:
    - Open a terminal and run sudo su to get root privileges.
    - Use nmap -T4 -A -v www.moviescope.com to scan for open ports and services, server details, and other relevant information.
5. Banner Grabbing:
    - Establish a Telnet connection using telnet www.moviescope.com 80.
    - Type GET / HTTP/1.0 to obtain server banners which reveal the make, model, and version of the web server software.

What I Learned:

- Web Reconnaissance: Understanding and using tools to gather detailed information about a target web application.
- Whois and DNS Interrogation: Collecting domain and server information to map out the web infrastructure.
- Port Scanning and Banner Grabbing: Identifying open ports and services, and extracting server details.

Task 2: Perform Web Application Reconnaissance using WhatWeb

Steps:

1. Log into Parrot Security Machine:
    - Open the terminal and log in with root privileges (sudo su).
2. Run WhatWeb:
    - Execute whatweb www.moviescope.com to identify web technologies used by the target site.
    - Run whatweb -v www.moviescope.com for a detailed report, including IP address, plugins, and HTTP header information.
3. Export Results:
    - Use whatweb --log-verbose=MovieScope_Report www.moviescope.com to save the results to a text file.
    - Open the file with pluma MovieScope_Report.

What I Learned:

- WhatWeb Tool: Using WhatWeb for identifying web technologies and gathering detailed information about the target site's infrastructure.
- Detailed Reporting: Generating and exporting detailed reconnaissance reports.

Task 3: Perform Web Spidering using OWASP ZAP

Steps:

1. Launch OWASP ZAP:
    - Open the terminal and start OWASP ZAP (zaproxy).

2. Automated Scan:
    - Enter the target URL (www.moviescope.com) and start the automated scan.
    - Observe the scan results, focusing on the Spider tab for web spidering details.
3. Review Results:
    - Check the Spider tab for URLs and hidden content.
    - Review detailed information under the Messages tab.

What I Learned:

- Web Spidering: Using OWASP ZAP to discover hidden content and functionality within a web application.
- Vulnerability Identification: Understanding how spidering aids in identifying potential vulnerabilities.

Task 4: Detect Load Balancers using Various Tools

Steps:

1. Run Dig Command:
    - Use dig yahoo.com to identify DNS load balancers.
2. Use LBD Tool:
    - Run lbd yahoo.com to detect DNS and HTTP load balancing.

What I Learned:

- Load Balancer Detection: Identifying load balancers that distribute traffic to increase web application reliability.
- Tools for Detection: Using dig and lbd for detecting load balancers and analyzing their configurations.

Task 5: Identify Web Server Directories using Various Tools

Steps:

1. Use Nmap:
    - Run nmap -sV --script=http-enum www.moviescope.com to enumerate web server directories.

2. Use Gobuster:
    - Copy the wordlist file common.txt and use it with Gobuster: gobuster dir -u www.moviescope.com -w /home/attacker/Desktop/common.txt.
3. Use Dirsearch:
    - Navigate to the dirsearch directory and run python3 dirsearch.py -u http://www.moviescope.com to identify directories.
    - Use additional options for specific file extensions and excluding status codes.

What I Learned:

- Directory Enumeration: Identifying exposed directories and files on web servers.
- Tools Utilization: Using Nmap, Gobuster, and Dirsearch for comprehensive directory enumeration.

Task 6: Perform Web Application Vulnerability Scanning using Vega

Steps:

1. Launch Vega:
    - Switch to Windows Server 2022 and ensure WampServer is running.
    - Switch to Windows 11, launch Vega, and start a new scan on http://10.10.1.22:8080/dvwa.
2. Configure Scan:
    - Select all Injection and Response Processing Modules.
    - Initiate and monitor the scan, reviewing the discovered vulnerabilities under Scan Alerts.

What I Learned:

- Vulnerability Scanning: Using Vega to detect security issues in web applications.

- Comprehensive Scanning: Configuring scans to identify a wide range of vulnerabilities, including SQL Injection and XSS.

Task 7: Identify Clickjacking Vulnerability using ClickjackPoc

Steps:

1. Create Domain File:
    - Navigate to the ClickjackPoc directory and create domain.txt with the target URL.
2. Run ClickjackPoc:
    - Execute python3 clickJackPoc.py -f domain.txt to start the scan.
3. Verify Vulnerability:
    - Open the generated HTML file in Firefox to verify the clickjacking vulnerability.

What I Learned:

- Clickjacking Detection: Identifying vulnerabilities where user interactions can be hijacked.
- Tool Usage: Using ClickjackPoc to perform clickjacking tests and verify the presence of vulnerabilities.

Lab 02: Perform Web Application Attacks

Task 1: Perform a Brute-force Attack using Burp Suite

Steps:

1. Launch WampServer in Windows Server 2022:
    - Switch to the Windows Server 2022 machine.
    - Activate the machine and log in with the default credentials (CEH\Administrator, Pa$$w0rd).

- Search for "wampserver64" in the search field and launch WampServer64.
- Wait for the WampServer icon to turn green, indicating that it's successfully running.

2. Access WordPress Login Page:
    - Switch to the Parrot Security machine.
    - Launch the Mozilla Firefox browser.
    - Navigate to the WordPress login page: http://10.10.1.22:8080/CEH/wp-login.php?
3. Configure Burp Suite Proxy:
    - Open Mozilla Firefox, go to Preferences > Network Settings.
    - Select manual proxy configuration.
    - Set HTTP Proxy to 127.0.0.1 and Port to 8080.
    - Check "Also use this proxy for FTP and HTTPS" and click OK.
4. Launch Burp Suite:
    - Open Burp Suite from the Applications menu.
    - Enter the password "toor" if prompted.
    - Accept the terms and conditions.
5. Configure Burp Suite:
    - Select the Temporary project and click Next.
    - Choose "Use Burp defaults" and click Start Burp.
6. Initiate Burp Suite Proxy:
    - Switch to the Proxy tab in Burp Suite.
    - Ensure interception is on (Intercept is on).
7. Initiate Brute-force Attack:
    - Enter random credentials in the WordPress login page (e.g., admin, password).
    - Switch back to Burp Suite and intercept the HTTP request.

- Right-click the intercepted request and choose "Send to Intruder".

8. Configure Intruder:
    - Switch to the Intruder tab.
    - Select the "Cluster bomb" attack type.
    - Clear default payload values in the Positions tab.
    - Set username and password as payload values.
    - Load username and password wordlist files.
9. Launch Attack:
    - Start the attack by clicking the "Start attack" button.
10. Analyze Results:
    - Monitor the progress of the attack.
    - Note down successful username-password combinations.
11. Turn off Intercept:
    - Once done, switch back to the Proxy tab.
    - Turn off interception (Intercept is off).
12. Test Successful Credentials:
    - Remove the proxy settings in Firefox.
    - Reload the WordPress login page.
    - Log in with the obtained credentials.

What I Learned:

- Brute-force Attack: Using Burp Suite to automate the process of guessing credentials.
- Proxy Configuration: Configuring Burp Suite as a proxy to intercept and manipulate web traffic.
- Payload Configuration: Setting up payloads and launching attacks using Burp Suite's Intruder tool.

Task 2: Perform Parameter Tampering using Burp Suite

Steps:

1. **Access Target Website:**
    - Launch Mozilla Firefox in the Parrot Security machine.
    - Navigate to the target website: www.moviescope.com.
2. **Configure Burp Suite Proxy:**
    - Open Mozilla Firefox, go to Preferences > Network Settings.
    - Select manual proxy configuration.
    - Set HTTP Proxy to 127.0.0.1 and Port to 8080.
    - Check "Also use this proxy for FTP and HTTPS" and click OK.
3. **Launch Burp Suite:**
    - Open Burp Suite from the Applications menu.
    - Enter the password "toor" if prompted.
    - Accept the terms and conditions.
4. **Configure Burp Suite:**
    - Select the Temporary project and click Next.
    - Choose "Use Burp defaults" and click Start Burp.
5. **Initiate Burp Suite Proxy:**
    - Switch to the Proxy tab in Burp Suite.
    - Ensure interception is on (Intercept is on).
6. **Login and Intercept Request:**
    - Log in to the target website with valid credentials.
    - Switch back to Burp Suite and intercept the HTTP request.
7. **Manipulate Parameters:**
    - Right-click the intercepted request and choose "Send to Repeater".
    - In Repeater, modify the parameters (e.g., change ID values) and observe the response.
8. **Test Parameter Tampering:**

- Experiment with different parameter values to observe changes in the application's behavior.

9. **Turn off Intercept:**
    - Once done, switch back to the Proxy tab.
    - Turn off interception (Intercept is off).

What I Learned:

- **Parameter Tampering:** Manipulating parameters exchanged between the client and server to modify application data.
- **Proxy Usage:** Configuring Burp Suite as a proxy to intercept and modify web requests and responses.
- **Repeater Tool:** Using Burp Suite's Repeater tool to manually modify and resend intercepted requests for testing.

Task 3: Identify XSS Vulnerabilities in Web Applications using PwnXSS

Steps:

1. **Launch Terminal:**
    - Open the Parrot Terminal from the Applications menu.
2. **Switch to Root User:**
    - Type **sudo su** and enter the root password ("toor").
3. **Navigate to PwnXSS Directory:**
    - Change directory to the PwnXSS directory.
4. **Run PwnXSS Scanner:**
    - Execute **python3 pwnxss.py -u http://testphp.vulnweb.com** to scan the target website for XSS vulnerabilities.
5. **Analyze Results:**
    - Review the output to identify any detected XSS vulnerabilities.
6. **Exploit Vulnerabilities:**
    - Copy any detected XSS link from the terminal output.
7. **Test XSS Payload:**

- Open Mozilla Firefox and paste the copied XSS link into the address bar.
- Observe the behavior of the target website to confirm XSS vulnerability.

What I Learned:

- **XSS Vulnerability Detection:** Using PwnXSS to identify cross-site scripting vulnerabilities in web applications.
- **Payload Injection:** Exploiting XSS vulnerabilities by injecting malicious scripts into web pages.
- **Security Assessment:** Conducting security scans to identify and mitigate potential security risks in web applications.

Task 4: Exploit Parameter Tampering and XSS Vulnerabilities in Web Applications

Parameter Tampering:

1. Access Target Website:
   - Launch Mozilla Firefox and navigate to www.moviescope.com.
   - Log in with username "steve" and password "password".
   - Click on the View Profile tab to access the profile page.
2. Parameter Tampering:
   - Note the value of ID in the address bar (e.g., ID=4).
   - Change the parameter in the address bar to ID=1 and observe the change in profile (e.g., profile of "sam").
   - Similarly, try different ID values to access profiles of different users without hacking techniques.

XSS Vulnerability:

3. Exploiting XSS:
   - Click on the Contacts tab.
   - Enter any name (e.g., "steve") in the Name field.

- Inject a cross-site script in the Comment field.
- Click Submit Comment.

4. Test XSS Payload:
    - Refresh the Contacts page.
    - If prompted to resend information, click Resend.
    - Observe the execution of the malicious script, indicating XSS vulnerability.

What I Learned:

- Parameter tampering allows accessing different profiles by modifying parameters in the URL.
- XSS vulnerability enables injecting and executing malicious scripts on web pages.

Task 5: Perform Cross-site Request Forgery (CSRF) Attack

1. Access Target WordPress Website:
    - Open a web browser and navigate to http://10.10.1.22:8080/CEH/wp-login.php.
    - Log in with username "admin" and password "qwerty@123".
2. Exploit CSRF Vulnerability:
    - Activate the vulnerable plugin "leenk.me".
    - Enable Facebook module with specific settings.
    - Obtain API Token from the website's Edit Profile page.
3. Execute CSRF Attack:
    - Use WPScan with obtained API Token to enumerate vulnerable plugins.
    - Identify and exploit CSRF vulnerability in the "leenk.me" plugin.
    - Copy the provided security script to a network-shared folder.
4. Launch CSRF Attack:

- Access the shared folder from the Windows Server 2022 machine.
- Copy and paste the security script to the desktop.
- Open the script with Firefox.

5. Result:
    - Successful execution of the CSRF attack, altering settings on the target website.

What I Learned:

- CSRF attack exploited the vulnerability in the "leenk.me" plugin to manipulate website settings.
- Demonstrated the potential impact of CSRF attacks on web applications.

Task 6: Enumerate and Hack a Web Application using WPScan and Metasploit

1. Launch WampServer:
    - Switch to the Windows Server 2022 machine.
    - Activate WampServer by clicking on it in the system tray.
2. Enumerate Usernames with WPScan:
    - Switch to the Parrot Security machine.
    - Open a terminal window and switch to the root user.
    - Start the PostgreSQL service if necessary.
    - Launch WPScan with the API token and target URL to enumerate usernames.
    - Note down the identified usernames.
3. Crack Passwords with Metasploit:
    - Launch Metasploit Framework from the terminal.
    - Use the wordpress_login_enum auxiliary module.
    - Set options including PASS_FILE, RHOSTS, RPORT, TARGETURI, and USERNAME.

- Execute the module to perform a dictionary attack and crack passwords.
- Note down the cracked password for the targeted user.

4. Log in to WordPress Website:
    - Open Firefox and navigate to the WordPress login page.
    - Log in with the obtained username and cracked password.
    - Confirm successful login and access to the WordPress website content.
5. Repeat for Other User Accounts:
    - Repeat the process to crack passwords for other identified usernames.
    - Log in to the WordPress website with each cracked password to verify access.

What I Learned:

- Utilizing WPScan to enumerate usernames on a WordPress website.
- Cracking passwords using Metasploit's auxiliary module.
- Gaining unauthorized access to a WordPress website by exploiting weak credentials.

Task 7: Exploit a Remote Command Execution Vulnerability to Compromise a Target Web Server

1. Launch WampServer:
    - Ensure WampServer is running on the Windows Server 2022 machine.
2. Access Damn Vulnerable Web App (DVWA):
    - Switch to the Windows 11 machine.
    - Open Firefox and navigate to the DVWA login page.
    - Log in with provided credentials.
3. Exploit Command Injection Vulnerability:
    - Navigate to the Command Injection page in DVWA.

- Attempt ping and command injection to test vulnerabilities.
- Change security level to low to exploit command execution vulnerability.

4. Extract Information and Perform Commands:
    - Use command injection to extract system information (e.g., hostname, running processes).
    - Execute commands like taskkill to manipulate processes.
    - View directory structure, user accounts, and add a new user with administrative privileges.
5. Access Target Machine via Remote Desktop Connection:
    - Log in to the target Windows Server 2022 machine using the newly created administrator account.
    - Establish a remote desktop connection using Remote Desktop Connection tool.
    - Verify successful login and access to the target machine.

What I Learned:

- Exploiting command execution vulnerabilities in web applications like DVWA.
- Extracting system information and manipulating processes remotely.
- Adding user accounts with administrative privileges to compromise the target system.

Task 8: Exploit File Upload Vulnerability at Different Security Levels

Step-by-Step Procedure:

1. **Generate Payload**:
    - Use **msfvenom** to generate a raw payload (**php/meterpreter/reverse_tcp**) with specific listener settings (**LHOST** and **LPORT**).
    - Copy the generated payload to the clipboard.

2. **Create PHP File**:
    - Navigate to the Desktop and open a new file named **upload.php** using the Pluma text editor.
    - Paste the copied payload into the file and save it.
3. **Set Up Listener**:
    - Open a Terminal window and launch **msfconsole**.
    - Set up a listener using the **exploit/multi/handler** module with appropriate payload and listener settings.
    - Start the listener.
4. **Upload Payload** (Low Security Level):
    - Access the DVWA website, log in as admin/password, and set the security level to Low.
    - Navigate to the File Upload page and upload the **upload.php** file.
    - Confirm successful upload and note the file location.
5. **Execute Payload**:
    - Access the uploaded **upload.php** file in the browser to trigger the payload execution.
    - Observe the successful establishment of a Meterpreter session in the Terminal.
6. **Repeat for Medium and High Security Levels**:
    - Repeat steps 1 to 5, adjusting security levels and payload settings accordingly.

Key Learnings:

- **Understanding Payload Generation**: Learning how to generate payloads using **msfvenom** with specific settings.
- **Exploiting File Upload Vulnerabilities**: Identifying and exploiting file upload vulnerabilities in web applications.

- **Setting Up Listeners**: Configuring listeners in Metasploit to capture incoming connections from exploited vulnerabilities.
- **Interacting with Meterpreter**: Initiating and interacting with Meterpreter sessions to gain control over target systems.

Task 9: Gain Access by Exploiting Log4j Vulnerability

Step-by-Step Procedure:

1. **Setup Vulnerable Server**:
    - Switch to the Ubuntu machine and deploy a Docker container hosting a Log4j vulnerable server.
2. **Prepare Exploitation**:
    - Install JDK 8 and update JDK path in the exploitation script (**poc.py**).
    - Open a netcat listener on the Parrot Security machine.
3. **Generate Payload**:
    - Run the exploitation script (**poc.py**) with appropriate parameters to create and execute the payload.
    - Copy the generated payload from the script output.
4. **Execute Payload**:
    - Access the Log4j vulnerable server in the browser and paste the payload into the Username field.
    - Submit the payload and observe the reverse shell connection in the netcat listener.

Key Learnings:

- **Simulating Vulnerable Environments**: Setting up and deploying vulnerable applications for testing and exploitation.
- **Custom Scripting for Exploitation**: Learning how to modify and run custom scripts (**poc.py**) to automate exploitation.
- **Exploiting Log4j Vulnerability**: Understanding how to leverage Log4j vulnerability (CVE-2021-44228) to gain unauthorized access.

- **Reverse Shell Handling**: Initiating and handling reverse shell connections to establish backdoor access to target systems.