
PROJECT REPORT - PHASE 2

TOPIC

“Implementation of the Security Program: Vulnerability Analysis and Penetration Testing”

CYBR 7910: CAPSTONE IN CYBERSECURITY PRACTICUM

KENNESAW STATE
UNIVERSITY

PREPARED BY

- Uju Anachebe
- Masrura Tasnim
- Gevorges Chenmala
- Richard Beiswanger

Kennesaw State University

PREPARED FOR

Zhigang Li, Ph.D.

Assistant Professor,
Department of Information Systems
College of Computing and SWE

Kennesaw State University

April 14, 2024

Table of Content

1.	STATUS REPORT	1
2.	INTRODUCTION	1
3.	IMPLEMENTATION OF THE SECURITY PROGRAM	2
3.1	OVERVIEW OF SECURITY TOOLS AND CONFIGURATIONS	2
3.2	INSTALLATION AND CONFIGURATION OF SECURITY TOOLS	4
4.	VULNERABILITY ANALYSIS AND PENETRATION TESTING STRATEGY	9
5.	REFERENCES.....	11



KENNESAW STATE
UNIVERSITY

1. Status Report

As we transitioned from phase 1 to phase 2 of our project, we began to implement the plans detailed in the previous report. This involved exploring Akwaaba's webserver. We started by setting up proper credentials for server and WordPress access, noting all existing software and configurations for the operating system, database, and WordPress, and updating these existing elements to prevent exploits of outdated software. Next, we installed the security tools on the server and installed WordPress security plugins. These were configured to operate in the environment while allowing WordPress to operate as intended. Finally, we selected vulnerability analysis and penetration testing tools that would be used for Phase 3, and practiced using them on our web server to find out what additional changes we should make to better secure the system in preparation for the final phase of this project.

We used the timetable as a guideline and held regular daily meetings to help each other learn and operate the tools, installations, and configurations and conduct security operations. There were some tools that, after trial and error, proved not to work in our specific environment, so we selected alternatives listed in detail in the following report. Our original Gant Chart had deliverables completed a week ahead of time, but we decided to change this plan for this section for two reasons: 1. The next phase relies on working with other teams to run our analysis and tests on their server as they conduct their tests on ours, and there is no way to begin these tasks before the actual deadline for this second phase. 2. The time necessary to install and learn the tools and troubleshoot the various configuration issues was grossly underestimated. We were glad to have this extra time to iron out all our issues as we completed these tasks to a satisfactory level. Going forward, we still intend to complete the third phase in one week, and we will put in all the man-hours necessary to accomplish this goal. This plan has given us the flexibility to allocate our time to the tasks that need it most, rather than being limited by an inflexible schedule.

2. Introduction

The cybersecurity landscape has been constantly evolving, requiring a hands-on approach to learning and defense. In this capstone project, we engage in practical security implementation and testing through a structured learning experience. After the completion of Phase One, this report aims to outline a detailed explanation of Phase 2. This phase comprises two primary objectives: the implementation of a robust security program on an assigned virtual machine (VM) and the execution of vulnerability analysis and penetration testing against another team's VM.

Our team implements a comprehensive security program on a designated virtual machine (VM), using firewalls, antivirus software, and intrusion detection systems. Through these

activities, we assess the VM's specific needs and implement security measures reflecting current best practices. Following, our team conducts vulnerability analysis and penetration testing on our VM. This exercise evaluates the effectiveness of security implementations and helps us understand how vulnerabilities are exploited.

Our goal is to evaluate multiple security tools such as Nikto, Gobuster, Nmap, and WPScan. We will also implement a vulnerability analysis strategy with multiple testing applications such as Nmap, OpenVAS, and Metasploit Framework.

3. Implementation of the Security Program

3.1 Overview of Security Tools and Configurations

In the second phase of our project, the primary objective of implementing a security program is to establish a robust defensive protocol for the assigned virtual machine (VM). With this, we can ensure its capabilities in handling common cyber threats and vulnerabilities. We selected a number of security tools and configurations based on a comprehensive review of tool documentation and training materials. Our goal is to create a secure IT (Information Technology) environment that prevents unauthorized access and reduces security risks. We also considered the following in designing our study:

- **Prevention:** Our first step is to explore the system for software and configurations needing updates and establishing auto-updates where possible. We also wanted to actively prevent security breaches by introducing strong defensive mechanisms that eliminate potential attacks.
- **Detection:** We selected numerous tools including to detect unauthorized access and security breaches that could compromise the WordPress server's integrity.
- **Response:** We wanted tools that allowed us to respond effectively to security incidents, minimizing their impact and restoring normal operations as quickly as possible.
- **Compliance:** We selected tools that followed established security rules, to ensure all security measures comply with cybersecurity standards and best practices, maintaining the confidentiality, integrity, and availability of the data stored and processed on our WordPress Server.

The configuration and deployment of each tool within the VM's security infrastructure provides an understanding as well as proper implementation of said security tools. This strategy is designed to prevent any vulnerabilities and cyber threats against our assigned VM. The security measures we implemented include Nikto, Gobuster, and WPScan:

- The first major issue in configuration we found while examining the VM, is that WordPress and its plugins could not update. After much troubleshooting, we finally found a solution in adjusting the SELinux (Security-Enhanced Linux) Policy. SELinux is a strong security architecture included in Linux, but in this case, it was blocking necessary actions by FPM. FPM is an important packaging management tool. After reviewing the audit.log we edited the SELinux policy to allow the actions necessary for WordPress and its plugins to successfully communicate with their servers.
- Let'sEncrypt: Next, we ensured our WordPress site utilized https for higher security. We attempted to use Let'sEncrypt to generate valid Certificates so we could use https, but we found certificates from a 3rd party is not possible without a domain name for our server. Our solution was to generate self-signed certificates by installing mod_ssl. To conclude https protection, we set up virtualhosts so that all http traffic automatically forwarded to https.
- Firewall Configuration (UFW - Uncomplicated Firewall): This firewall offers a first line of defense by controlling incoming and outgoing network traffic based on predetermined security rules. The firewall plays a critical role in shielding the VM from network-based threats by limiting connections to essential services and blocking all other unauthorized access attempts.
- Fail2Ban: A service that strictly watches login attempts, its main purpose is to prevent brute force attacks. After installation we set the limits so frequent login attempts over a low threshold, along with multiple failed attempts would ban users for 24 hours or up to a week, preventing any ability to brute force passwords.
- ModSecurity: ModSecurity acts as a smart firewall with more security features utilizing a vast cybersecurity rule engine. ModSecurity proved to need careful configuration, as by default it doesn't like IP addresses in the URL or automatic forwarding, two things we decided are necessary for our situation. Fortunately, like SELinux, ModSecurity also has a toggle switch in its config file for troubleshooting during setup.
- Httpd.conf required reconfiguration as well. This file sets up the overall rules for http navigation, and there were initial issues with the configuration making the site less secure. We adjusted the Directory block, DocumentRoot, DocumentRoot Directory, and switched the Indexes and FollowSymLinks Options to be more secure.
- Web Server Scanner (Nikto): We selected Nikto to perform comprehensive tests against web servers, detecting outdated software, harmful files, and other potential vulnerabilities.
- Directory Enumeration Tool (Gobuster): We considered Gobuster as a tool to enumerate directories and file locations on web servers.

- Antivirus Software (ClamAV): ClamAV detects, classifies, and eliminates malicious software, trojans, and viruses. Regular updates to its database ensure it remains effective against new variants of malware.
- Wazuh: A platform for real-time threat detection, log analysis, and compliance monitoring. We installed this tool as a single-node docker deployment on the server itself so that we could monitor attacks in real time during the next phase of our project.

Each of these tools is configured with specific rules and policies that reflect the unique needs and risk factors associated with the VM. Our goal is to create a security framework tailored to the specific operational environment. With this strategy, we hope to enhance security while ensuring effective operation of the WordPress server.

3.2 Installation and Configuration of Security Tools

- Nikto (Tests Against Web Servers)
 - Installation process:
Step #1: We installed Nikto on Kali Linux. Nikto is a Perl-based security tool and its installation on Kali Linux is straightforward as it is included in Kali's package repository. We installed Nikto with the following command:

```
sudo apt install nikto
```

After the installation, we confirmed whether or not Nikto is correctly installed by running 'nikto -h' command to see a help menu that lists out the command-line options. We also ran the following command that initiates Nikto to perform a scan against the target host with our VM's IP address.

```
nikto -h <IP ADDRESS>
```

Step #2: We then update Nikto Before proceeding with the scan. Maintaining an updated plugin database is crucial for effective vulnerability scanning. Before conducting a scan, the documentation recommends updating Nikto's plugin database using the following command:

```
nikto -update
```

Step #3: After updating the database, we repeat the initial scan command to leverage the latest vulnerability checks.

Step #4: Nikto can also scan websites secured with HTTPS protocol. Use the -ssl option as shown below, replacing <IP ADDRESS> with our VM's IP address:

```
nikto -h https://<IP ADDRESS> -ssl
```

Step #5: To save the scan results in an HTML report file named "report.html", we use the -output option along with the desired HTML filename:

```
nikto -h <IP ADDRESS> -output report.html
```

Step #6: By default, Nikto does not follow redirects encountered during the scan. As a result, to scan the destination after a redirect, we use the -following redirects command:

```
nikto -h <IP ADDRESS> -followredirects
```



```
kali@kali:~$ nikto -h 10.96.33.211 -output report.html
- Nikto v2.5.0

+ Target IP: 10.96.33.211
+ Target Hostname: 10.96.33.211
+ Target Port: 80
+ Start Time: 2024-04-13 16:06:48 (GMT-4)

+ Server: Apache/2.4.37 (Red Hat Enterprise Linux) OpenSSL/1.1.1k
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /6x1MPqJf.php: Retrieved x-powered-by header: PHP/8.0.30.
+ OpenSSL/1.1.1k appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ Apache/2.4.37 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ Hostname '10.96.33.211' does not match certificate's names: akwaaba. See: https://cwe.mitre.org/data/definitions/297.html
+ OPTIONS: Allowed HTTP Methods: OPTIONS, HEAD, GET, POST, TRACE .
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /home/: Drupal Link header found with value: <https://10.96.33.211/home/index.php?rest_route=/>; rel="https://api.w.org/". See: https://www.drupal.org/
+ /home/: This might be interesting.
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ 8909 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time: 2024-04-13 12:23:57 (GMT-4) (553 seconds)

+ 1 host(s) tested

(kali@kali)~$ nikto -h 10.96.33.211 -output report.html
- Nikto v2.5.0

+ Target IP: 10.96.33.211
+ Target Hostname: 10.96.33.211
+ Target Port: 80
+ Start Time: 2024-04-13 16:06:48 (GMT-4)

+ Server: Apache/2.4.37 (Red Hat Enterprise Linux) OpenSSL/1.1.1k
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://10.96.33.211/
+ No CGI Directories found (use "-C all" to force check all possible dirs)
+ Apache/2.4.37 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OpenSSL/1.1.1k appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 19 error(s) and 4 item(s) reported on remote host
+ End Time: 2024-04-13 16:07:04 (GMT-4) (16 seconds)

+ 1 host(s) tested
```

Step #7: For more detailed output during the scan, including additional information about discovered vulnerabilities, we use the -Display V option:

```
nikto -h <IP ADDRESS> -followredirects -Display V
```

```
kali@kali -
File Actions Edit View Help
ViSat Apr 13 17:04:26 2024 - 302 for GET: /96.tgz
ViSat Apr 13 17:04:26 2024 - 302 for GET: /96.tgz
ViSat Apr 13 17:04:26 2024 - 302 for GET: /10.96.33.211.war
ViSat Apr 13 17:04:27 2024 - 302 for GET: /10.96.33.211.war
ViSat Apr 13 17:04:27 2024 - 302 for GET: /dump.zip
ViSat Apr 13 17:04:27 2024 - 302 for GET: /dump.zip
ViSat Apr 13 17:04:27 2024 - 302 for GET: /site.sql
ViSat Apr 13 17:04:27 2024 - Running scan for "SSL and cert checks" plugin
ViSat Apr 13 17:04:27 2024 - Running scan for "HTTP Options" plugin
ViSat Apr 13 17:04:27 2024 - 200 for OPTIONS: /
ViSat Apr 13 17:04:27 2024 - 302 for OPTIONS: /
ViSat Apr 13 17:04:27 2024 - 302 for WAOJIBK: /
ViSat Apr 13 17:04:27 2024 - 302 for DEBUG: /
ViSat Apr 13 17:04:27 2024 - 400 for PROPFIND: /
ViSat Apr 13 17:04:27 2024 - 302 for TRACE: /
ViSat Apr 13 17:04:27 2024 - 302 for TRACE: /
ViSat Apr 13 17:04:27 2024 - 302 for TRACE: /
ViSat Apr 13 17:04:27 2024 - 302 for TRACE: /
ViSat Apr 13 17:04:27 2024 - Running scan for "Server Messages" plugin
ViSat Apr 13 17:04:27 2024 - Running scan for "strutshock" plugin
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - Running scan for "dishwasher" plugin
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - Running scan for "shellshock" plugin
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:27 2024 - for GET:
ViSat Apr 13 17:04:28 2024 - for GET:
ViSat Apr 13 17:04:28 2024 - for GET:
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 18 error(s) and 2 item(s) reported on remote host
+ End Time: 2024-04-13 17:04:28 (GMT-4) (23 seconds)

+ 1 host(s) tested
ViSat Apr 13 17:04:28 2024 + 1609 requests made in 23 seconds

(kali@kali)-[~]
└─$
```

- Configuration details and rationale: Nikto comes pre-configured with various tests that are designed to detect common vulnerabilities and configurations. However, we can update Nikto's configuration file (nikto.conf) to scan our website with our specific needs in mind.
- Gobuster (discover hidden URLs, files, and directories)
 - Installation process: Gobuster is a Go-based tool used for URL brute-forcing and it is included in the Kali Linux repositories. We installed Gobuster by running the following command:

`sudo apt update -y`

We then check for upgradable packages with the following command:

`apt list --upgradable`

Then we install Golang with the following:

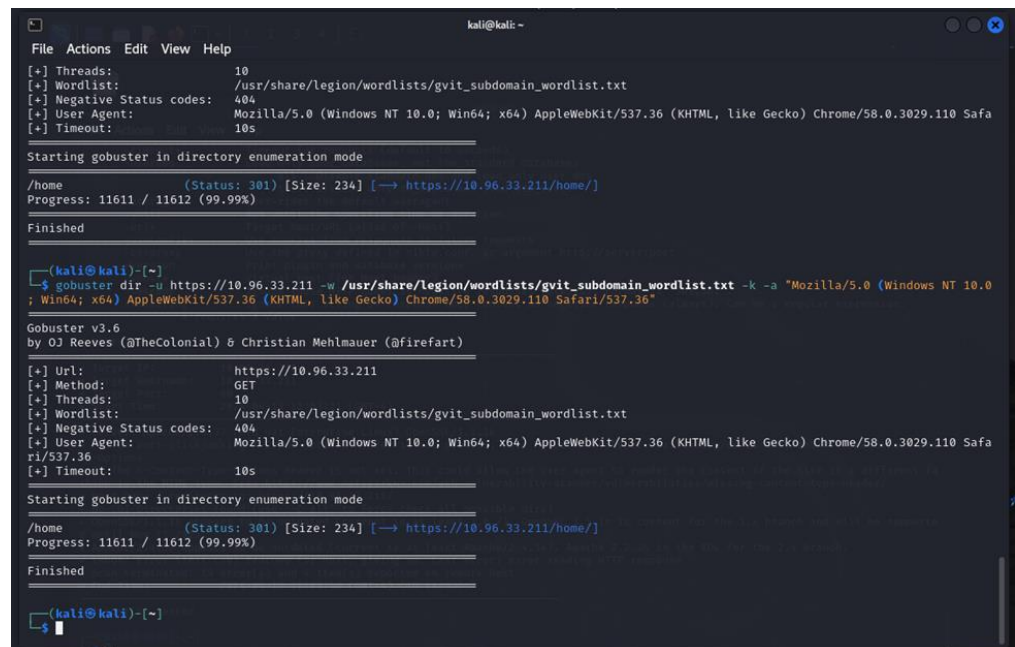
`sudo apt install golang -y`

Then we install Gobuster using the following command:


```
sudo apt install gobuster -y
```

To verify the successful installation, we run 'gobuster -help' command in the terminal which displays the usage and options for the tool. Gobuster operates in different modes, with the dir mode used for directory enumeration. To use Gobuster more effectively, we configure it with the following:

- A comprehensive wordlist provided by Kali Linux in /usr/share/wordlists/. This allows Gobuster to attempt a wide range of directories and file names. (gobuster dir -u <target URL> -w <wordlist>)
- Concurrency settings to control how many threads are used, balancing speed against the risk of overloading the web server.
- Extensions that are relevant to our web applications such as .php, .html, .js, etc. which allow us to find specific types of resources or backup files that might be exposed unintentionally.
- Status code exclusion to filter out responses that do not indicate a successful find, thereby streamlining the results.



```
kali@kali: ~  
File Actions Edit View Help  
[+] Threads: 10  
[+] Wordlist: /usr/share/legion/wordlists/gvit_subdomain_wordlist.txt  
[+] Negative Status codes: 404  
[+] User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36  
[+] Timeout: 10s  
Starting gobuster in directory enumeration mode  
/home (Status: 301) [Size: 234] [→ https://10.96.33.211/home/]  
Progress: 11611 / 11612 (99.99%)  
Finished  
  
(kali@kali) ~  
$ gobuster dir -u https://10.96.33.211 -w /usr/share/legion/wordlists/gvit_subdomain_wordlist.txt -k -a "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"  
  
Gobuster v3.6  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
[+] Url: https://10.96.33.211  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: /usr/share/legion/wordlists/gvit_subdomain_wordlist.txt  
[+] Negative Status codes: 404  
[+] User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36  
[+] Timeout: 10s  
Starting gobuster in directory enumeration mode  
/home (Status: 301) [Size: 234] [→ https://10.96.33.211/home/]  
Progress: 11611 / 11612 (99.99%)  
Finished  
  
(kali@kali) ~  
$
```

This command scans the web server at https://<IP ADDRESS> for directories using the wordlist /usr/share/legion/wordlists/gvit_subdomain_wordlist.txt:

```
gobuster      dir      -u      https://<IP      ADDRESS>      -w
/usr/share/legion/wordlists/gvit_subdomain_wordlist.txt
```

- Configuration details and rationale: The reason for these configurations is to perform a methodical and efficient search for resources that could be exploited if they were not meant to be publicly accessible. The chosen settings aim to strike a balance between thoroughness and efficiency, ensuring that the tool provides actionable intelligence without excessive demands on the web server's resources.
- WPScan (vulnerability database for WordPress)
 - WPScan was already preinstalled in Kali Linux. After running an update command, we used WPScan on the IP address with this command: `wpscan --url https://<IP ADDRESS>/home -e u,vp --disable-tls-checks --ignore-main-redirect --stealthy`

```

kali-linux-2024.1-virtualbox-amd64 [F]
kali@kali: ~
File Actions Edit View Help
kali@kali: ~
kali@kali: ~$ wpscan --url https://10.96.33.211/home -e u,vp --disable-tls-checks --ignore-main-redirect --stealthy
WPScan
WordPress Security Scanner by the WPScan Team
Version 3.8.0
Sponsored by Automattic - https://automattic.com/
@WPScan, @ethicalhack3r, @erwan_l, @irefart

[-] URL: https://10.96.33.211/home/ [10.96.33.211]
[-] Started: Sat Apr 13 10:08:31 2024

Interesting Finding(s):

[-] Headers
  - Server: Apache/2.4.37 (Red Hat Enterprise Linux) OpenSSL/1.1.1k
  - Powered By: PHP/8.0.29
  Found By: Headers (Passive Detection)
  Confidence: 100%

[-] XML-RPC seems to be enabled: https://10.96.33.211/home/xmlrpc.php
  Found By: Link Tag (Passive Detection)
  Confidence: 100%
  References:
  - http://codex.wordpress.org/XML-RPC-Pingback_API
  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[-] WordPress version 6.5.2 identified (latest, released on 2024-04-09).
  Found By: WSS Generator (Passive Detection)
  - https://10.96.33.211/home/feed-rss2_generatorhttps://wordpress.org/7v-e.5.2/generators
  - https://10.96.33.211/home/feed-comments-rss2_generatorhttps://wordpress.org/7v-e.5.2/generators

[-] WordPress theme in use: dyad
  Location: https://10.96.33.211/home/wp-content/themes/dyad/
  Latest Version: 1.0.10 (up to date)
  Last Updated: 2024-02-08 09:00:00
  Style URL: https://10.96.33.211/home/wp-content/themes/dyad/style.css?ver=6.5.2
  Style Name: Dyad
  Style URI: https://wordpress.com/themes/dyad/
  Description: dyad pairs your written content and images together in perfect balance. The theme is geared towards ...
  Author: Automattic
  Author URI: http://wordpress.com/themes
  Found By: CSS Style in Homepage (Passive Detection)
  Version: 1.0.10 (80% confidence)
  Found By: Style (Passive Detection)
  - https://10.96.33.211/home/wp-content/themes/dyad/style.css?ver=6.5.2, Match: "version: 1.0.10"

[-] Enumerating Vulnerable Plugins (via Passive Methods)
[-] Checking Plugin Versions (via Passive Methods)
[-] No plugins Found.

[-] Enumerating Users (via Passive Methods)
[-] User(s) Identified:

[-] Known(s):
  Found By: Author Posts - Display Name (Passive Detection)
  Confirmed By: WSS Generator (Passive Detection)
  No WPScan API token given, as a result vulnerability data has not been output.
  You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sat Apr 13 10:08:44 2024
[+] Requests Done: 9
[+] Cached Requests: 4
[+] Data Sent: 2.358 KB
[+] Data Received: 239.798 KB
[+] Memory used: 254.051 MB
[+] Elapsed time: 00:00:12

```

WPScan is an open-source security tool that is used to find vulnerabilities that can be found on WordPress websites. It primarily scans for vulnerabilities in WordPress' plugins and themes. It also explores other issues that could be exploited by attackers. WPScan will be used to scan for any vulnerabilities that could potentially be on our server. The security tool provided details concerning the

headers, themes, plugins, and enumerated users. It showed the version of WordPress and the Dyad theme is currently up to date. The tool detected the presence of XML_RPC with a 30% confidence level suggesting a potential risk if it is not properly configured or used. No vulnerable plugins were found at the time, and WPScan enumerated a username (ksuitg5).

4. Vulnerability Analysis and Penetration Testing Strategy

- Nmap (Network Scanning)
 - Nmap, is an open-source network scanning tool used by network administrators, security professionals, and ethical hackers. It can discover network topologies, perform security audits, and assess vulnerabilities. Nmap allows us to explore networks to identify hosts and open ports and discover information about services running on the target machine.
 - We selected this tool for ease of use in identifying various services running on the server.
- OpenVAS (Vulnerability Scanning)
 - OpenVAS, an open vulnerability assessment scanner, is utilized for detecting and controlling vulnerabilities found in systems and networks. Primarily, the security tool detects security flaws within servers by scanning for known vulnerabilities, improper configurations, and entry points for attackers. The scanner can perform comprehensive scans on servers and networks with the use of a database that contains several known vulnerabilities. OpenVAS can perform local and remote security checks. The security tool provides detailed reports that users can use to address their security issues effectively.
 - We selected this tool because of its extensive vulnerability database that provides accurate security findings. OpenVAS is also frequently updated with the most recent threat intelligence of known security vulnerabilities. It even allows us to customize the scans to meet our specific requirements. We selected this tool because it produces well-rounded reports that aid in making decisions.
- Metasploit Framework (Penetration Testing)
 - Metasploit is an open-source penetration testing tool. It provides a collection of exploit modules that can be configured to target known vulnerabilities on a server. We can conduct security assessments, exploit vulnerabilities, and select options to integrate with other tools.
 - We selected this tool because it has an extensive list of modules that can run vulnerabilities against websites that are powered by WordPress.

- OWASP ZAP (Web Vulnerability Scanner)
 - OWASP ZAP is an open-source security tool that identifies security vulnerabilities in web applications. This security tool can intercept and modify HTTP requests, allowing us to quickly identify security issues such as SQL injections or broken authentication. It provides a comprehensive scanner to analyze all vulnerabilities found on the web application. With this security tool we can generate reports detailing vulnerabilities and recommended solutions.
 - We selected this tool because of its well-rounded features that allow us to successfully perform security scans. This security tool is regularly developed and kept up to date with current security trends and vulnerabilities. This web vulnerability scanner will aid in securing our web application from evolving threats.
- XSSer
 - XSSer is a tool to check crosssite scripting vulnerabilities. This tool can automate the posting of malicious code through HTTP GET and POST request on a webpage that accepts user submitted text message contents.
 - We selected this tool because the Akwaaba website has a feature to accept customer posted messages on their site.
- Burp Suite
 - Burp Suite is a suite of security tools included with Kali Linux. Its main objective is to capture HTTP requests which can be used in other security tools, or to edit and resend the HTTP request. Burp Suite will start tracing the site traffic which is watched from the history tab. It can also intercept traffic allowing you to block or manipulate it and forward the request to the server.
 - Our rationale for choosing this tool is because it allows us to intercept traffic that can be analyzed by many other security tools like XSSer.

6. References

- Burp Suite - Application Security Testing Software - PortSwigger. (n.d.). Retrieved April 13, 2024, from <https://portswigger.net/burp>
- Cisco. (n.d.). ClamAVNet. Retrieved April 13, 2024, from <https://www.clamav.net/>
- Fail2ban / fail2ban. GitHub. (2022). Retrieved April 13, 2024, from <https://github.com/fail2ban/fail2ban>
- Let's Encrypt. (n.d.). Retrieved April 13, 2024, from <https://letsencrypt.org/>
- Mebus/cup: Common User Passwords Profiler (CUPP) - GitHub. (n.d.). Retrieved April 13, 2024, from <https://github.com/Mebus/cupp>
- Metasploit - Penetration Testing Tool - Rapid7. (n.d.). Retrieved April 13, 2024, from <https://www.rapid7.com/products/metasploit/>
- Nikto 2.5 | CIRT.net. (n.d.). Retrieved April 13, 2024, from <https://cirt.net/Nikto2>
- Nmap / nmap. GitHub. (n.d.). Retrieved April 13, 2024, from <https://github.com/nmap/nmap>
- OJ/gobuster: Directory/File, DNS and VHost busting tool written in Go - GitHub. (n.d.). Retrieved April 13, 2024, from <https://github.com/OJ/gobuster>
- Owasp-Modsecurity/Modsecurity. GitHub. (2024). Retrieved April 13, 2024, from <https://github.com/owasp-modsecurity/ModSecurity>
- Sqlmap: automatic SQL injection and database takeover tool. (n.d.). Retrieved April 13, 2024, from <https://sqlmap.org/>
- Wazuh/wazuh: Wazuh - The Open Source Security Platform - GitHub. Unified XDR and SIEM protection for endpoints and cloud workloads. (n.d.). Retrieved April 13, 2024, from <https://github.com/wazuh/wazuh>
- Wordfence security – firewall, malware scan, and login security. WordPress. (n.d.) Retrieved April 13, 2024 from <https://wordpress.org/plugins/wordfence/>
- WPSCAN CLI Scanner. WPScan. (2023, October 20). Retrieved April 13, 2024, from <https://wpscan.com/wordpress-cli-scanner/>
- XSSer: Cross Site “Scripter.” (n.d.). Retrieved April 13, 2024, from <https://xsser.03c8.net/>