Portfolio Lab on WPScan for Penetration Testing of a WordPress Site

This lab will guide you through setting up a WordPress security assessment lab using WPScan on a Kali Linux virtual machine (VM) and a WordPress VM. You will learn how to configure the environment, install the necessary tools, and use WPScan to explore different scanning options against a WordPress installation.

WPScan is a powerful tool used to scan WordPress websites for security vulnerabilities, themes, plugins, and configuration weaknesses.

Primary Learning Outcome

LO3: Evaluate risks to privacy and anonymity in commonly used applications.

Lab Setup Requirements

Prerequisites

- **VirtualBox** installed on your machine.
- Two VMs:
    1. **Kali Linux VM**: For running WPScan and performing security assessments.
    2. **WordPress VM**: An instance with WordPress installed, configured as a target for scanning.

Step 1: Setting Up the Kali Linux VM

1. **Get WPScan API Key**:
    - Register for a free WPScan API key to access the vulnerability database from https://wpscan.com/register.
    - Save the API key; you'll use it in the scanning commands.

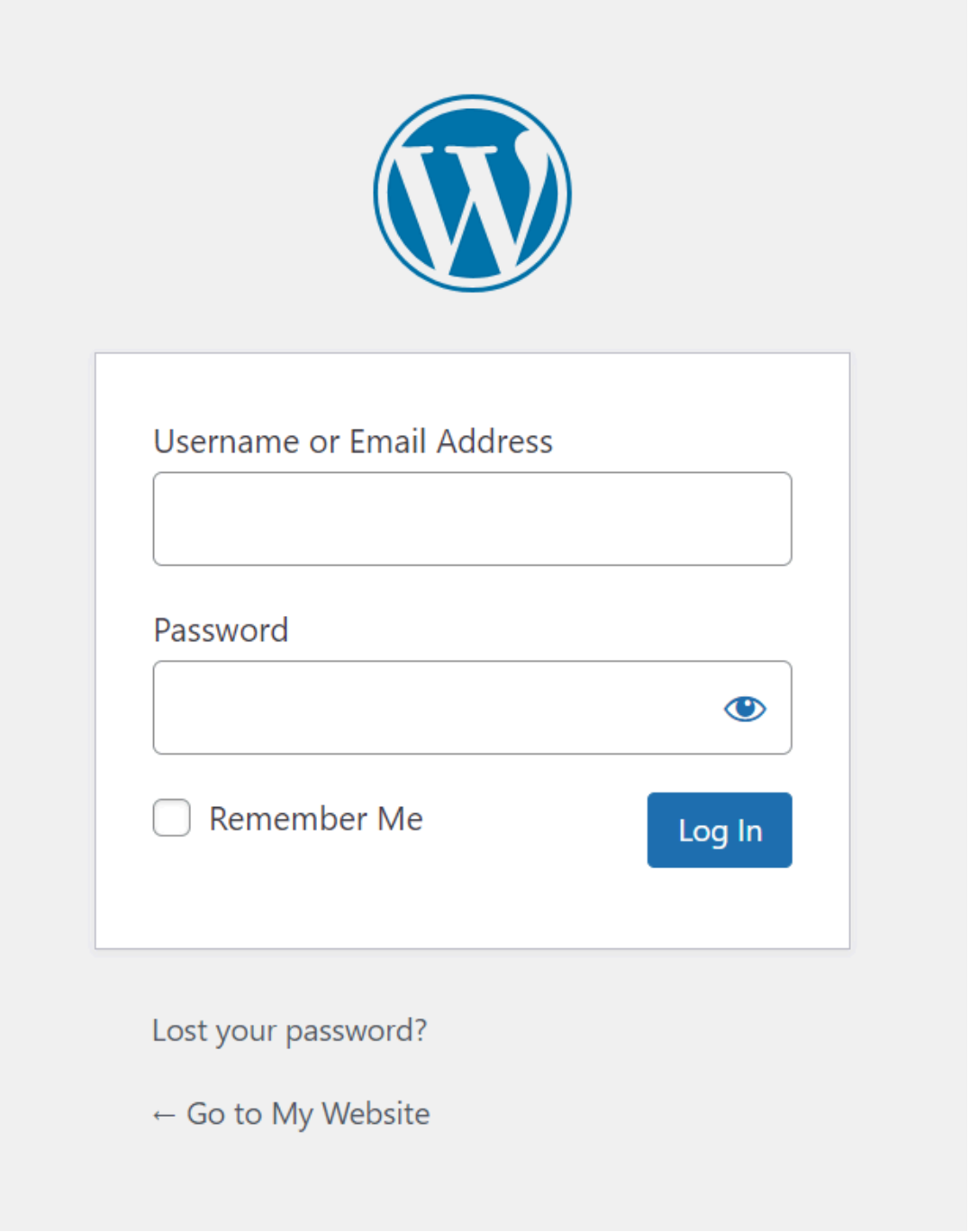
Step 2: Setting Up the WordPress VM

1. Download a Preconfigured WordPress VM from the Cyber Lab VM Repository
    - Download the preconfigured WordPress VM from Bitnami or create one manually using a Linux server image and installing WordPress.
    - Assign the VM at least 1 GB of RAM and 10 GB storage.

Step 3: Ensure that the Kali VM can communicate with the WordPress VM

1. **Start both VMs**:
    - Make sure that both virtual machines can ping each other successfully
2. **Make sure that your Kali virtual machine can browse to the WordPress virtual machine.** :
    - Use the `ifconfig` , `ip addr` or `ip a` command on the WordPress VM to determine its IP address if required.
    - From the Kali virtual machine, open a web browser and enter the IP address of your WordPress virtual machine.
3. **Browse to the admin pages of WordPress and create a user with a weak password.** :
    - From the Kali virtual machine open a web browser and enter the IP address of the WordPress virtual machine, appended with /wp-admin. For example: **192.168.123.xxx/wp-admin**

- A page similar to that shown below should appear:



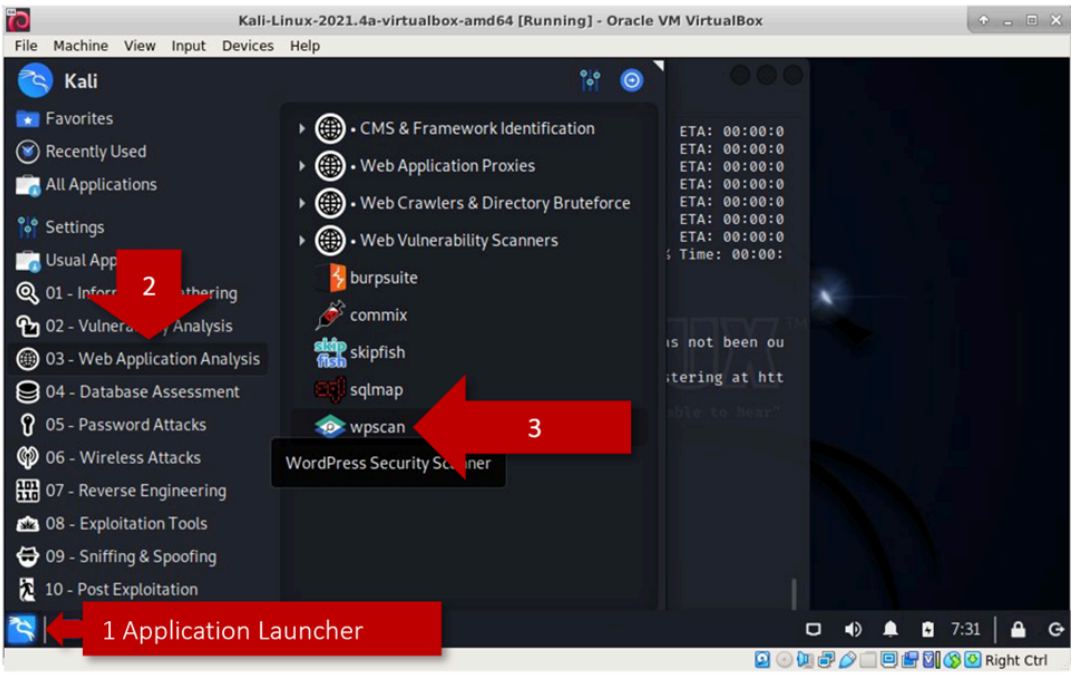
- Login and create a new user with admin privileges. Ensure that the new user has a weak password. For example, you can try using a password from some examples here: https://en.wikipedia.org/wiki/List_of_the_most_common_passwords
- Once you have created the new user, test it!!!

Step 4: Can you add a vulnerable theme and plugin to WordPress?

1. Research vulnerable WordPress themes. See if you can add a vulnerable theme to WordPress.
2. Research vulnerable WordPress plugins. See if you can add a vulnerable plugin to WordPress.

Step 5: Update WPScan

1. On your Kali virtual machine, select the Kali Applications Launcher > select 03 – Web Application Analysis > wpscan as indicated in the image below.



This will open a new terminal window. From the wpscan terminal type:

```
wpscan --update
```

WPScan Lab Exercises

1. Vulnerability Database Update and Plugin Enumeration

Before a vulnerability scan, update the WPScan vulnerability database using your API key:

```
wpscan --api-token YOUR_API_KEY --update
wpscan --url http://<wordpress_ip> --enumerate p
```

- **Explanation**: WPScan checks its vulnerability database for the latest issues and then enumerates installed plugins (`p`). Knowing the plugins helps identify possible attack vectors since plugins often have vulnerabilities.

2. Full Scan for Plugins, Themes, and WordPress Version Detection

Perform a comprehensive scan that targets plugins, themes, and the WordPress version.

```
wpscan --url http://<wordpress_ip> --enumerate vp,vt,tt
```

- **Explanation**:
  - `vp`: Enumerates vulnerable plugins.
  - `vt`: Enumerates vulnerable themes.

- `tt`: Enumerates themes and attempts to find vulnerabilities associated with them.
- This scan is more thorough and can take some time but provides essential information about potential vulnerabilities in plugins, themes, and the core WordPress version.

3. Brute Forcing Login Passwords

If usernames are discovered, you can perform a brute-force password attack. **(Ensure you have authorisation to perform this on the target system.)**

1. Download a file called ***10000_common_passwords.txt*** from the Cyber Lab VM Repository. You will need to get this file on to your Kali VM.
2. Run the brute-force scan with the following command:

```
wpscan --url http://<wordpress_ip> --passwords passwords.txt --username
<username>
```

- **Explanation**: This command will attempt to log in to WordPress with a list of passwords for the given username. Be mindful that brute-forcing can be slow, depending on the password list's length.

4. Scanning with API Key for Detailed Vulnerabilities

To find detailed vulnerability information, use your WPScan API key:

```
wpscan --api-token YOUR_API_KEY --url http://<wordpress_ip> --enumerate ap
```

- **Explanation**: The `ap` option enumerates all plugins and checks each plugin against the WPScan vulnerability database. Using the API key enables access to the latest vulnerability information.

5. Custom User-Agent and Proxy Options

To simulate real-world attack scenarios, try using a custom User-Agent string or a proxy.

```
wpscan --url http://<wordpress_ip> --user-agent "Mozilla/5.0 (Windows NT 10.0;
Win64; x64)" --proxy http://<proxy_ip>:<proxy_port>
```

- **Expected Outcome of Point 6: Using a Custom User-Agent and Proxy in WPScan**

  In point 6, the WPScan command is configured to simulate real-world scanning conditions by:

  1. **Setting a Custom User-Agent String**
  2. **Routing Traffic Through a Proxy Server**

  Here is an overview of what this command does.

---

1. Custom User-Agent String

- **Purpose**: The custom User-Agent string (`Mozilla/5.0 (Windows NT 10.0; Win64; x64)`) makes WPScan requests look like they're coming from a regular web browser (in this case, a Windows 10 machine with a 64-bit version of Firefox or Chrome).
- **Expected Outcome**:

- **Avoid Detection**: This setting helps bypass basic detection mechanisms that block scans by identifying typical User-Agent strings associated with tools like WPScan.
- **Bypass WAF Rules**: Many web application firewalls (WAFs) block or restrict automated tool access based on specific User-Agent signatures. By changing it, you might bypass some WAF rules designed to detect and block security scanners.
- **More Authentic Simulation**: Scanning with a User-Agent that mimics regular traffic makes the scan look like typical user behavior, which can be useful for testing realistic security responses.

---

2. Proxy Configuration

- **Purpose**: The `--proxy` option routes all WPScan traffic through an intermediary proxy server. This setup can be used to hide the origin IP address or to monitor the network traffic during scanning.
- **Expected Outcome**:
  - **Anonymity and IP Masking**: If you're testing a site where IP-based detection or blocking is in place, using a proxy can help mask your IP, making it harder for the target site to detect your true location or network origin.
  - **Rate-Limiting Evasion**: Some sites rate-limit requests by IP address. By using a proxy, you might avoid these rate limits, especially if you rotate proxies.
  - **Traffic Inspection**: Using a proxy allows you to capture and analyse WPScan requests and responses in real time. This is helpful if you want to review or troubleshoot how WPScan interacts with the target site.
  - **Testing Against External Systems**: Some organisations configure security systems differently for internal and external requests. By using an external proxy, you can test how the WordPress site responds to external IP addresses versus those within an internal network.

Combining a custom User-Agent and proxy, WPScan requests are made to look more like ordinary user traffic while hiding the origin IP, making the scan harder to detect and potentially allowing for deeper analysis of how the WordPress site handles external requests. This setup can help evade basic IP-based or User-Agent-based blocks and provide insight into the site's real-world security responses to external scans.

6. Excluding Specific Scans and Setting a Delay

If you want to avoid certain types of scans or slow down the scanning process, use exclusion and delay options:

```
wpscan --url http://<wordpress_ip> --enumerate vp,vt --throttle 0.5
```

- **What is the aim of Point 7?**

  1. **Exclude Specific Types of Scans**
  2. **Throttle Requests by Adding a Delay**

  This setup is designed to achieve two main objectives:

---

1. Selective Scanning: Excluding Specific Scans

- **Aim**: By specifying only certain scan types (in this case, vulnerable plugins `vp` and themes `vt`), the scan becomes more targeted.
- **Benefits**:
  - **Focus on High-Risk Areas**: Limiting scans to plugins and themes focuses WPScan's efforts on the components most likely to have vulnerabilities in a WordPress installation. Plugins and themes often introduce vulnerabilities because they are frequently updated and sometimes contain exploitable code.
  - **Reduce Scan Time**: Scanning only for specific items (like plugins and themes) reduces the time needed to complete the scan, as it avoids unnecessary checks (such as user enumeration or core version detection).
  - **Limit System Load**: Fewer scan types mean fewer requests to the target WordPress site, which reduces the impact on server resources and network traffic.

2. Adding a Delay Between Requests

- **Aim**: The `--throttle` option (set to 0.5 seconds in this case) introduces a delay between each HTTP request sent by WPScan.
- **Benefits**:
  - **Avoid Detection**: Many security systems monitor the frequency of requests. Rapid-fire scanning is more likely to trigger intrusion detection/prevention systems or firewall rules. Adding a delay makes the scan appear more like regular user traffic.
  - **Prevent Rate-Limiting Blocks**: Some sites implement rate-limiting to prevent excessive requests in a short time. Introducing a delay helps avoid hitting these limits, allowing the scan to proceed without interruption.
  - **Reduce Server Strain**: Delaying requests is considerate to the target server, especially for production sites, as it reduces the load and helps maintain normal site performance during scanning.

Ultimately, the aim of point 7 is to perform a **stealthier, more targeted scan** by:

- Focusing only on the most critical parts of the WordPress site (plugins and themes) where vulnerabilities are most commonly found.
- Slowing down the request rate to reduce the likelihood of detection, avoid rate-limiting restrictions, and lower the impact on the target server.

This approach is particularly useful in real-world scenarios where maintaining a low profile is essential or when scanning a production site where minimising disruption is a priority.

Summary of WPScan Options

Command	Description
`--enumerate u`	Enumerate usernames.
`--enumerate p`	Enumerate plugins.
`--enumerate vp`	Enumerate vulnerable plugins.

Command	Description
`--enumerate vt`	Enumerate vulnerable themes.
`--enumerate tt`	Enumerate themes.
`--enumerate ap`	Enumerate all plugins.
`--passwords <file>`	Use a password list for brute-force attacks.
`--user-agent <string>`	Use a custom User-Agent string.
`--proxy <proxy_ip:proxy_port>`	Route traffic through a specified proxy.
`--throttle <seconds>`	Delay between requests.
`--api-token <api_key>`	Use API key to access vulnerability database.

Portfolio Deliverable

1. In-lab demonstration of Points 1 - 6
2. A brief report (no more than 300 words) with three headings and content under each heading. The headings are:
    - What vulnerability has been found (in either a theme or plugin)?
    - What are the risks associated with the vulnerability?
    - How the vulnerability can be mitigated?

End of Lab