



# Penetration Test Report Lab Week #10

Jordan Torres  
10-NOV-2024

Penetration Testing LLC

Upton II Room 366  
243 Centennial Dr Stop 7165  
Grand Forks, ND 58202  
United States of America

Tel: 1-701-777-2180  
Fax: We still use these :)  
Email: [jordan.torres@und.edu](mailto:jordan.torres@und.edu)  
Web: <http://www.penetrationtestingllc.com>

## Table of Contents

|  |                    |
|--|--------------------|
| <a href="#">Version Control.....</a>                                 | <a href="#">2</a>  |
| <a href="#">Point of Contact.....</a>                                | <a href="#">3</a>  |
| <a href="#">Contractor.....</a>                                      | <a href="#">3</a>  |
| <a href="#">Client.....</a>  | <a href="#">3</a>  |
| <a href="#">Project Details.....</a>                                 | <a href="#">4</a>  |
| <a href="#">Project Objectives.....</a>                              | <a href="#">4</a>  |
| <a href="#">Scope of Work.....</a>                                   | <a href="#">4</a>  |
| <a href="#">Period of Testing.....</a>                               | <a href="#">4</a>  |
| <a href="#">Executive Summary.....</a>                               | <a href="#">5</a>  |
| <a href="#">Exercise 2 - WPSCAN.....</a>                             | <a href="#">6</a>  |
| <a href="#">Exercise 3 - Web Application Scanning with WMAP.....</a> | <a href="#">11</a> |
| <a href="#">Conclusion.....</a>                                      | <a href="#">17</a> |
| <a href="#">Recommendations.....</a>                                 | <a href="#">17</a> |
| <a href="#">Risk Rating.....</a>                                     | <a href="#">18</a> |
| <a href="#">Reflection.....</a>                                      | <a href="#">19</a> |
| <a href="#">Appendix A: About Penetration Testing LLC.....</a>       | <a href="#">20</a> |

## Version Control

| Version | Title              | Author        | Description  | Date        |
|---------|--------------------|---------------|--|-------------|
| 0.1     | Initial Report     | Jordan Torres | Report Template Created                            | 10-OCT-2024 |
| 1.0     | Week 10 Lab Report | Jordan Torres | Created new report, screenshots, and edited report | 10-NOV-2024 |

## Point of Contact

### Contractor

Jordan Torres

Upton II Room 366  
243 Centennial Dr Stop 7165  
Grand Forks, ND 58202  
United States of America  
Email: [jordan.torres@und.edu](mailto:jordan.torres@und.edu)  
Tel: 1-701-777-2180

### Client

CSCI 487 Penetration Testing

## Project Details

### Project Objectives

Module 5, Exercise 3: Enumerate a Wordpress Site(APT)

Module 5, Exercise 4: Perform Web Application Scanning with WMAP (APT)

### Scope of Work

Complete the tasks in EC-Council's lab by working through module 5. After finishing, compile findings into a report and submit it for grading.

### Period of Testing

04-NOV-2024 to 10-NOV-2024

## Executive Summary

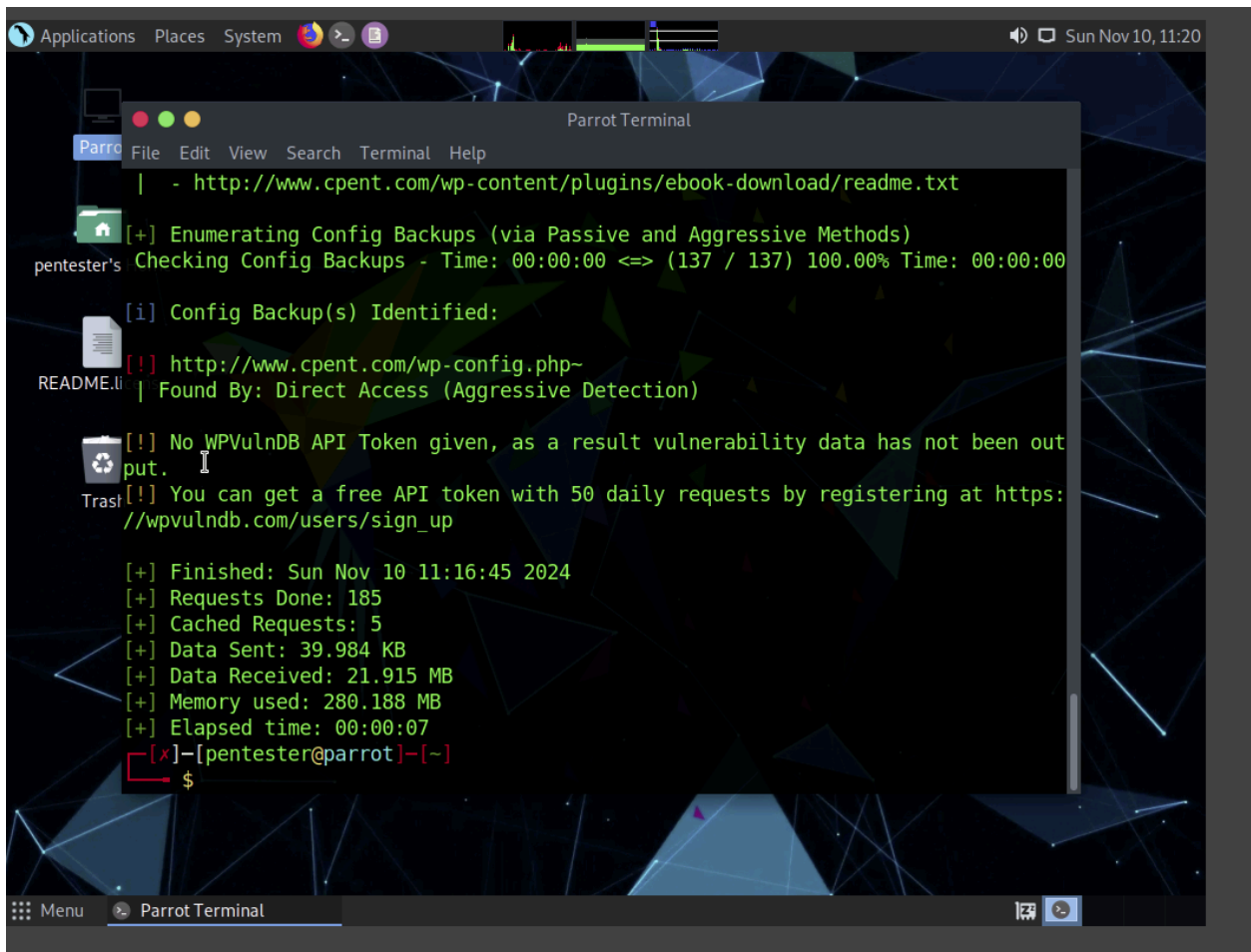
Developing skills in web vulnerability assessment using specific tools for WordPress enumeration and web application scanning. The wpscan tool was used to enumerate a WordPress site, enabling data collection on aspects such as plugins, themes, and user information. This exercise highlighted the importance of systematic enumeration, as following the outlined steps allowed data extraction regardless of the WordPress version. Also, introduced WMAP, a scanning tool within the Metasploit console, to scan web servers for vulnerabilities. Although WMAP has limitations compared to other web scanning tools like Nikto and Vega, it complements these tools by providing additional scanning options within the Metasploit framework. Together, these exercises underscored the value of employing multiple tools in web vulnerability assessment, as each tool has unique strengths that contribute to a well-rounded approach to identifying security gaps.

## Exercise 2 - WPSCAN

<https://www.kali.org/tools/wpscan/>

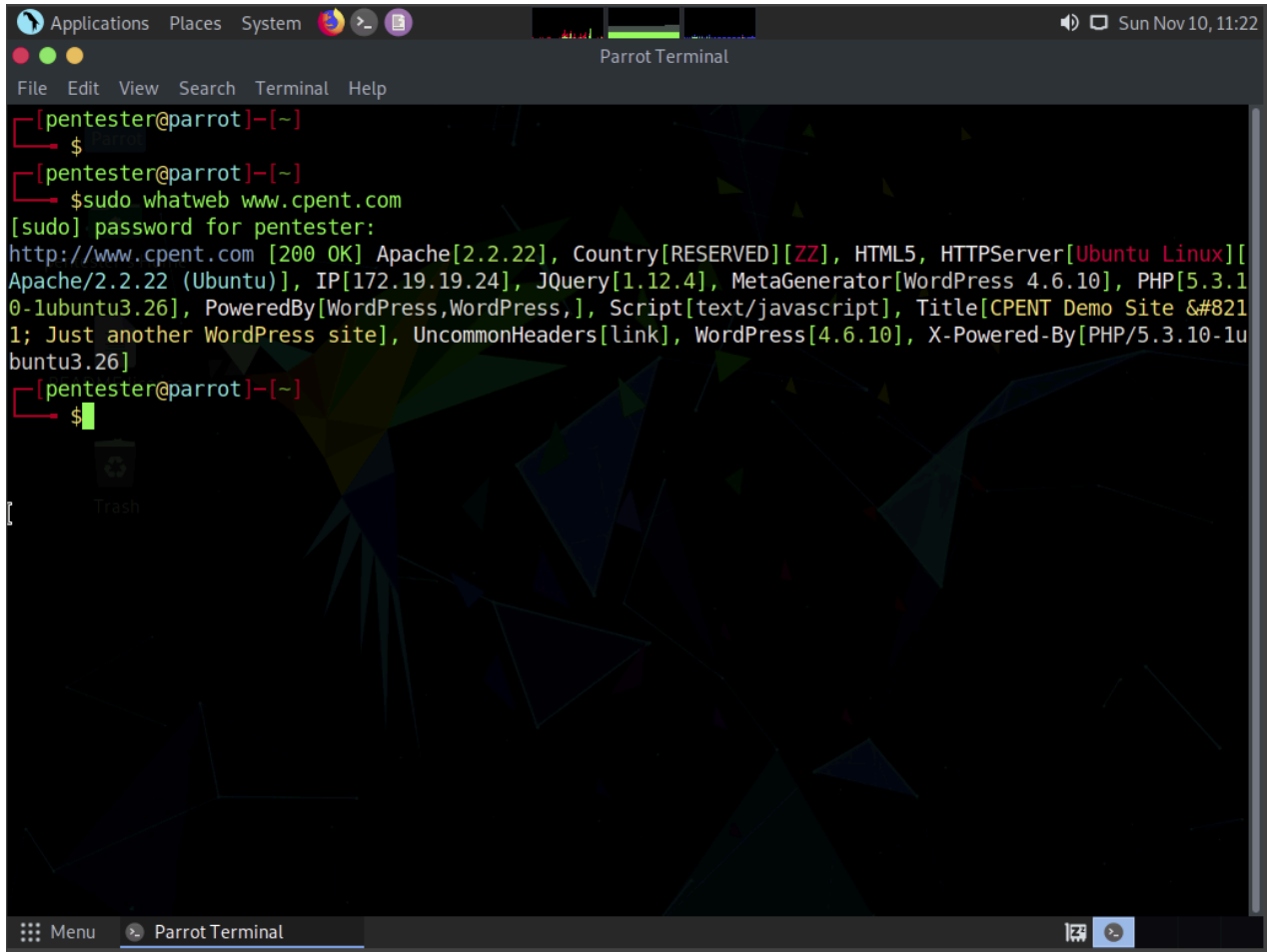
This scan will help check for vulnerabilities with wordpress.

Some screen shots of scanning



```
Parrot Terminal
File Edit View Search Terminal Help
| - http://www.cpent.com/wp-content/plugins/ebook-download/readme.txt
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 <=> (137 / 137) 100.00% Time: 00:00:00
[i] Config Backup(s) Identified:
[!] http://www.cpent.com/wp-config.php-
Found By: Direct Access (Aggressive Detection)
[!] No WPVulnDB API Token given, as a result vulnerability data has not been out
put.
[!] You can get a free API token with 50 daily requests by registering at https:
//wpvulndb.com/users/sign_up
[+] Finished: Sun Nov 10 11:16:45 2024
[+] Requests Done: 185
[+] Cached Requests: 5
[+] Data Sent: 39.984 KB
[+] Data Received: 21.915 MB
[+] Memory used: 280.188 MB
[+] Elapsed time: 00:00:07
[!] -[pentester@parrot]-[~]
$
```

Whatweb was excellent on showing the systems that are used, highly use for to find what vulnerabilities are used on specific versions.

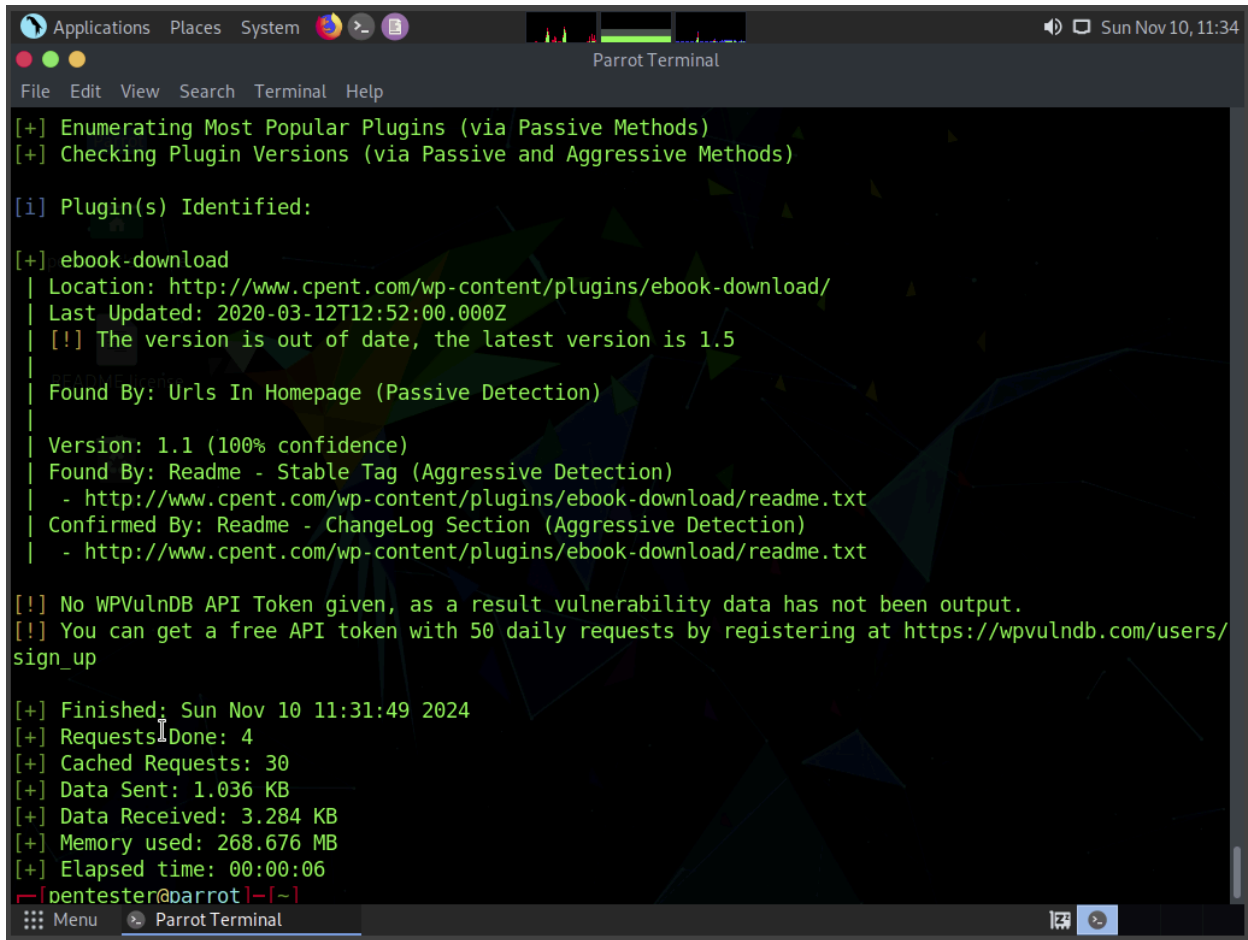


```
[pentester@parrot]~  
$  
[pentester@parrot]~  
$sudo whatweb www.cpent.com  
[sudo] password for pentester:  
http://www.cpent.com [200 OK] Apache[2.2.22], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][  
Apache/2.2.22 (Ubuntu)], IP[172.19.19.24], JQuery[1.12.4], MetaGenerator[WordPress 4.6.10], PHP[5.3.1  
0-lubuntu3.26], PoweredBy[WordPress,WordPress,], Script[text/javascript], Title[CPENT Demo Site &#821  
1; Just another WordPress site], UncommonHeaders[link], WordPress[4.6.10], X-Powered-By[PHP/5.3.10-lu  
buntu3.26]  
[pentester@parrot]~  
$
```



```
Applications Places System Parrot Terminal Sun Nov 10, 11:30
File Edit View Search Terminal Help
| Found By: Style (Passive Detection)
| - http://www.cpent.com/wp-content/themes/twentyfifteen/style.css?ver=4.6.10, Match: 'Version: 1.6'
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:01 <=====> (10 / 10) 100.00% Time: 00:00:01
[i] User(s) Identified:
[+] admin
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
[+] mike
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up
[+] Finished: Sun Nov 10 11:25:02 2024
[+] Requests Done: 26
[+] Cached Requests: 31
[+] Data Sent: 5.803 KB
[+] Data Received: 40.135 KB
[+] Memory used: 204.645 MB
[+] Elapsed time: 00:00:06
[~]pentester@parrot[~]
$
```

```
Applications Places System Parrot Terminal Sun Nov 10, 11:31
File Edit View Search Terminal Help
| Readme: http://www.cpent.com/wp-content/themes/twenty十六/readme.txt
| [!] The version is out of date, the latest version is 3.3
| Style URL: http://www.cpent.com/wp-content/themes/twenty十六/style.css
| Style Name: Twenty Sixteen
| Style URI: https://wordpress.org/themes/twenty十六/
| Description: Twenty Sixteen is a modernized take on an ever-popular WordPress layout – the horizon
tal masthead ...
| Author: the WordPress team
| Author URI: https://wordpress.org/
|
| Found By: Known Locations (Aggressive Detection)
| - http://www.cpent.com/wp-content/themes/twenty十六/, status: 500
|
| Version: 1.3 (80% confidence)
| Found By: Style (Passive Detection)
| - http://www.cpent.com/wp-content/themes/twenty十六/style.css, Match: 'Version: 1.3'
[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/
sign_up
[+] Finished: Sun Nov 10 11:30:38 2024
[+] Requests Done: 436
[+] Cached Requests: 16
[+] Data Sent: 103.278 KB
[+] Data Received: 467.547 KB
[+] Memory used: 214.582 MB
[+] Elapsed time: 00:00:05
[penetration@parrot]~$
```



```
Applications Places System > Parrot Terminal
File Edit View Search Terminal Help

[+] Enumerating Most Popular Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] ebook-download
| Location: http://www.cpent.com/wp-content/plugins/ebook-download/
| Last Updated: 2020-03-12T12:52:00.000Z
| [!] The version is out of date, the latest version is 1.5
| Found By: Urls In Homepage (Passive Detection)
| Version: 1.1 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|   - http://www.cpent.com/wp-content/plugins/ebook-download/readme.txt
| Confirmed By: Readme - Changelog Section (Aggressive Detection)
|   - http://www.cpent.com/wp-content/plugins/ebook-download/readme.txt

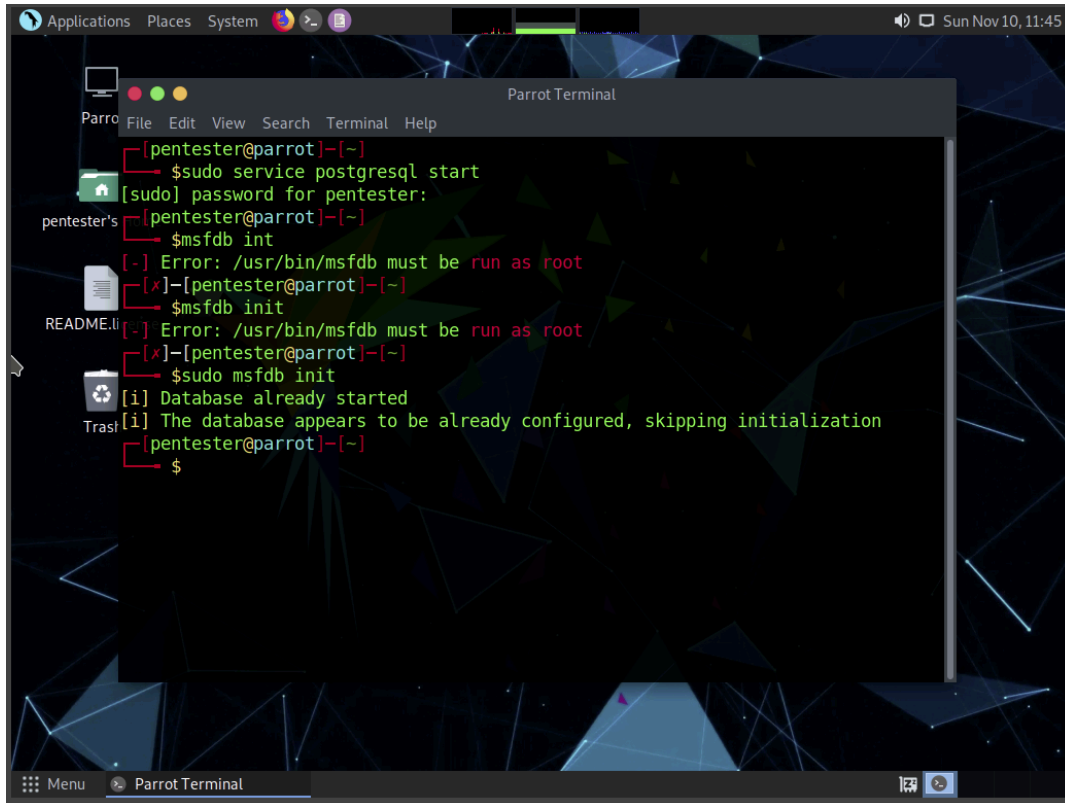
[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Sun Nov 10 11:31:49 2024
[+] Requests Done: 4
[+] Cached Requests: 30
[+] Data Sent: 1.036 KB
[+] Data Received: 3.284 KB
[+] Memory used: 268.676 MB
[+] Elapsed time: 00:00:06
pentester@parrot [~]
```

During this scan we are able to check on users, themes, and plugins. By checking on these vulnerabilities we can find more exploits.

## Exercise 3 - Web Application Scanning with WMAP

Scanned with Metasploit console



The screenshot shows a Parrot OS desktop with a dark, geometric background. A terminal window titled "Parrot Terminal" is open, displaying the following commands and output:

```
[pentester@parrot]~  
$sudo service postgresql start  
[sudo] password for pentester:  
[pentester@parrot]~  
$msfdb int  
[-] Error: /usr/bin/msfdb must be run as root  
[x]-[pentester@parrot]~  
$msfdb init  
[-] Error: /usr/bin/msfdb must be run as root  
[x]-[pentester@parrot]~  
$sudo msfdb init  
[i] Database already started  
[i] The database appears to be already configured, skipping initialization  
[pentester@parrot]~  
$
```

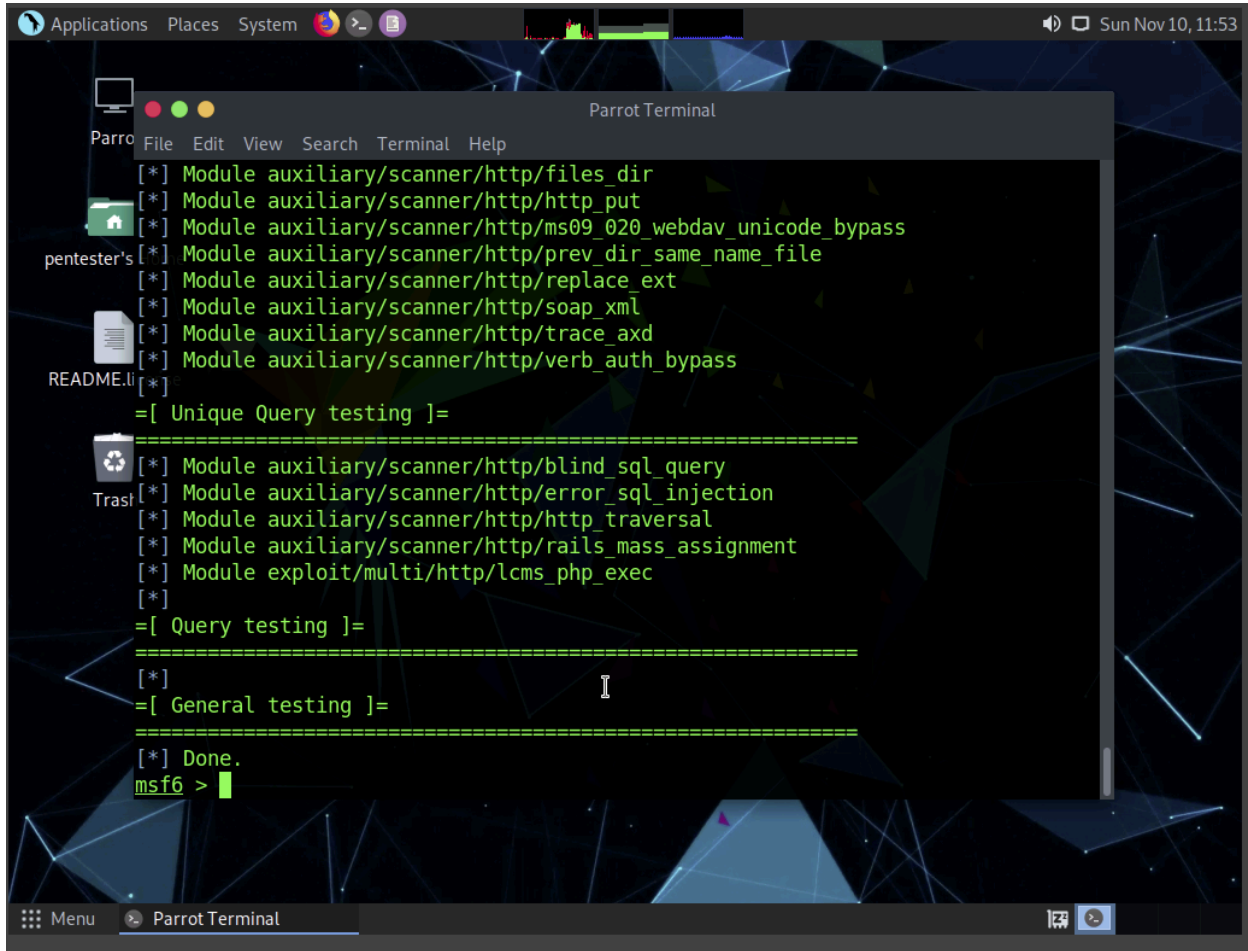
The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The desktop also shows a sidebar with icons for "Parrot", "pentester's", "README.li", and "Trash". The system tray at the bottom right shows the date and time: "Sun Nov 10, 11:45".

```
Parrot Terminal
File Edit View Search Terminal Help
k in run_single'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:468:in `each'
s/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:468:in `run_
single'
/usr/share/metasploit-framework/lib/rex/ui/text/shell.rb:158:in `run'
/usr/share/metasploit-framework/lib/metasploit/framework/command/console.rb:48:i
n `start'
l/usr/share/metasploit-framework/lib/metasploit/framework/command/base.rb:82:in `
start'
/usr/bin/msfconsole:23:in `<main>'
msf6 > wmap_targets -t http://172.19.19.22
msf6 > wmap_targets -t http://172.19.19.7
msf6 > wmap_targets -l
[*] Defined targets
=====

```

| Id | Vhost        | Host         | Port | SSL   | Path |
|----|--------------|--------------|------|-------|------|
| 0  | 172.19.19.22 | 172.19.19.22 | 80   | false | /    |
| 1  | 172.19.19.7  | 172.19.19.7  | 80   | false | /    |

```
msf6 > 
```



```
Parrot Terminal
File Edit View Search Terminal Help
[*] Module auxiliary/scanner/http/files_dir
[*] Module auxiliary/scanner/http/http_put
[*] Module auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
[*] Module auxiliary/scanner/http/prev_dir_same_name_file
[*] Module auxiliary/scanner/http/replace_ext
[*] Module auxiliary/scanner/http/soap_xml
[*] Module auxiliary/scanner/http/trace_axd
[*] Module auxiliary/scanner/http/verb_auth_bypass
[*]
=[ Unique Query testing ]=
=====
[*] Module auxiliary/scanner/http/blind_sql_query
[*] Module auxiliary/scanner/http/error_sql_injection
[*] Module auxiliary/scanner/http/http_traversal
[*] Module auxiliary/scanner/http/rails_mass_assignment
[*] Module exploit/multi/http/lcms_php_exec
[*]
=[ Query testing ]=
=====
[*]
=[ General testing ]=
=====
[*] Done.
msf6 >
```

```
Parrot Terminal
File Edit View Search Terminal Help

[*] Done.
msf6 > wmap_run -e
[*] Using ALL wmap enabled modules.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]   Site: 172.19.19.22 (172.19.19.22)
[*]   Port: 80 SSL: false

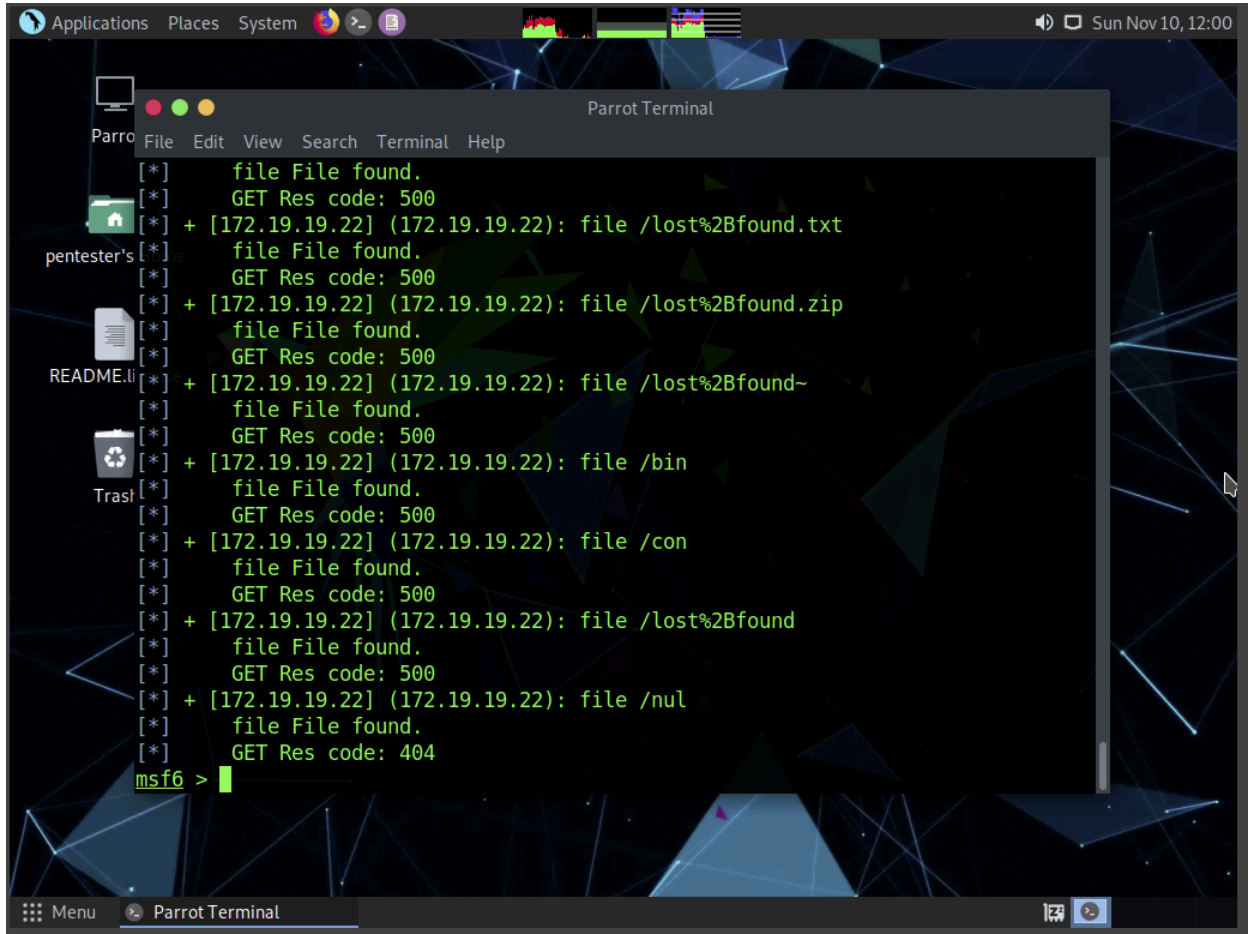
[*] Testing started. 2024-11-10 11:53:52 -0500
[*]
=[ SSL testing ]=

[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=

[*] Module auxiliary/scanner/http/http_version

[+] 172.19.19.22:80 Microsoft-IIS/8.5 ( Powered by ASP.NET, 500-Internal Server
Error )
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
```





Applications Places System Sun Nov 10, 12:00

Parrot Terminal

File Edit View Search Terminal Help

```
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /lost%2Bfound.txt
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /lost%2Bfound.zip
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /lost%2Bfound-
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /bin
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /con
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /lost%2Bfound
[*] file File found.
[*] GET Res code: 500
[*] + [172.19.19.22] (172.19.19.22): file /nul
[*] file File found.
[*] GET Res code: 404
msf6 >
```



```
Parrot Terminal
File Edit View Search Terminal Help
#####
No active nodes at this time
Stopping execution...
#####
No active nodes at this time
Stopping execution...
https://metasploit.com
No active nodes at this time
Stopping execution...
No active nodes at this time
+ -- ==[ metasploit v6.0.0-dev ]
+ -- ==[ 2052 exploits - 1108 auxiliary - 345 post ]
+ -- ==[ 566 payloads - 45 encoders - 10 nops ]
+ -- ==[ 7 evasion ]
No active nodes at this time
Metasploit tip: To save all commands executed since start up to a file, use the
makerc command
msf6 > vulns
Vulnerabilities
=====
Timestamp          Host          Name          References
-----
2024-11-10 17:09:39 UTC 172.19.19.7 HTTP Trace Method Allowed CVE-2005-3398,C
VE-2005-3498,OSVDB-877,BID-11604,BID-9506,BID-9561
```

<https://nvd.nist.gov/vuln/detail/CVE-2005-3398>

**CVE-2005-3398** affects Solaris versions 8, 9, and 10, where the default configuration of the Solaris Management Console (SMC) web server enables the HTTP TRACE method. This flaw could allow remote attackers to gain sensitive information, such as cookies and authentication details, from HTTP headers. The vulnerability is rated as medium risk and is classified under CWE-200 (Exposure of Sensitive Information).

<https://nvd.nist.gov/vuln/detail/CVE-2005-3498>

**CVE-2005-3398** vulnerability in IBM WebSphere Application Server versions 5.0.x, 5.1.x, and 6.x prior to specific fix versions. This issue occurs when session trace is enabled on the server. It causes the server to log the full URL, including query parameters, in the trace logs. This exposure of sensitive information could potentially allow attackers to access this data if they gain access to the trace logs.

## Conclusion

The versatility and thoroughness in scanning practices. By using wpscan for wordpress site enumeration and WMAP within Metasploit for web application scanning, the lab demonstrated how combining tools with unique strengths can enhance overall effectiveness in identifying security risks. I do remember years ago when we had to do all these by command line and it is nice to have these tools at our fingertips. Though each tool has its limitations, using them together builds a more robust security approach, equipping analysts with multiple perspectives for detecting and addressing vulnerabilities in web environments.

## Recommendations

Some recommendations

1. **Incorporate Additional Scanning Tools:** While wpscan and WMAP provide valuable insights, integrating other tools like Nikto, Vega, and OWASP ZAP could strengthen the assessment by identifying additional vulnerabilities and reducing blind spots.
2. **Regular Tool Updates and Version Checks:** Ensuring that tools like wpscan and WMAP are regularly updated will improve scanning accuracy and maintain compatibility with the latest security protocols. Updates often include enhanced detection capabilities and protection against newer vulnerabilities.
3. **Automate Scanning for Routine Checks:** Setting up automated scans on test environments can save time and provide consistent, up-to-date assessments. Automation is especially useful in larger web environments where regular, comprehensive scans may otherwise be time-consuming.
4. **Diversify Enumeration Approaches:** For WordPress enumeration, using other methods alongside wpscan—such as manual checks, plugin-specific scans, and login vulnerability testing—can help uncover additional security issues that might go unnoticed with a single tool.
5. **Analyze and Cross-Validate Results:** Comparing results across multiple tools can provide a more accurate view of vulnerabilities. Cross-validation helps distinguish between false positives and actual threats, leading to more reliable remediation efforts.

6. **Continued Training on Metasploit and Security Tools:** Given that Metasploit is a powerful framework with a wide range of capabilities, further training on its advanced modules and integrations with other security tools can help maximize its potential in real-world scenarios.

Implementing these recommendations would broaden assessment capabilities, enhance accuracy, and contribute to a more proactive and resilient approach to web security.

## Risk Rating

1. **Weak or Default WordPress Credentials**

- **Risk Rating:** **High**
- **Rationale:** Weak or default credentials are one of the most common attack vectors and could lead to full site compromise if exploited. Attackers can easily brute-force login credentials, gaining unauthorized access to the WordPress admin panel, which allows control over content, plugins, and potentially sensitive user data.

2. **Outdated WordPress Plugins and Themes**

- **Risk Rating:** **Medium to High**
- **Rationale:** Outdated plugins and themes are frequent sources of vulnerabilities, as they can contain unpatched security flaws. Attackers could exploit these to execute remote code or cross-site scripting (XSS) attacks. The severity depends on the specific vulnerability and whether a patch is available.

3. **Unsecured Web Application Ports**

- **Risk Rating:** **Medium**
- **Rationale:** Web applications with open, unsecured ports can expose the application to various attacks, such as unauthorized access or denial of service (DoS) attacks. While open ports are sometimes necessary, improper configurations or exposure of sensitive services can increase risk.

4. **Exposed Server Information**

- **Risk Rating:** **Low to Medium**
- **Rationale:** Information disclosure, such as server type, versions, and configurations, may seem low risk but can provide attackers with insights into potential weaknesses. By masking or hiding sensitive server information, the overall risk to the site can be reduced.

5. **Lack of SSL or Insecure HTTP Connections**

- **Risk Rating:** **High**
- **Rationale:** HTTP connections without SSL/TLS encryption expose user data to interception and compromise, especially login credentials and personal information. Using HTTPS reduces this risk significantly by encrypting data in transit.

## Reflection

These exercises highlighted the importance of combining technical skills with strategic thinking in web vulnerability assessment. Using wpscan and WMAP provided hands-on experience with each tool's unique functions, illustrating how targeted tools can identify common vulnerabilities in platforms like WordPress. Additionally, WMAP within Metasploit reinforced the idea of layered security, showing how different tools can complement each other for more comprehensive scans. This experience underscored the limitations of any single tool and the value of cross-validating results to better identify critical issues. Assigning risk ratings to vulnerabilities also deepened my understanding of prioritization, as risk assessment directly shapes remediation efforts. Ultimately, these exercises reinforced the need for a diverse cybersecurity toolkit and an adaptive approach to address evolving threats and technologies.

## Appendix A: About Penetration Testing LLC

Penetration Testing LLC is a specialized network security firm dedicated to safeguarding businesses from potential cyber threats. We offer comprehensive penetration testing services that identify vulnerabilities within networks, applications, and systems. Our team of certified ethical hackers utilizes cutting-edge tools and techniques to simulate real-world cyberattacks, helping organizations fortify their defenses. By delivering detailed reports and actionable recommendations, Penetration Testing LLC ensures that businesses remain resilient against emerging threats, while also maintaining compliance with industry standards and regulations. Our commitment to security excellence empowers clients to proactively protect their digital assets and maintain business continuity.