WEB APP TESTING SCRIPT MULTI-THREAD

Michael Awoyemi

CMP320: Advanced Ethical Hacking Unit 2

2023/24

.

Abstract

This report provides an overview of a Python script that automates multiple stages of Web application penetration testing. The script has been designed using the Open Worldwide Application Security Project methodology (OWASP). Its objective is to provide penetration testers with a faster and more efficient means of performing active reconnaissance scans with minimal input required. Once provided with the web server's IP address, the script assists testers in performing their tasks with increased speed and accuracy, which ultimately results in a more effective testing process.

The Python script is formed of several functions that run each scanning tool, parses the findings and generates the Word document report. It also uses multithreading as a means to execute every scan via independent threads which reduces the expected time of the output. Once the user has input the IP address and the scan is completed, the script generates several text files for each of the scans, which can be accessed by the tester for future reference. The tools used for the scanning phase are the WhatWeb scan, Nikto scan, Nmap scan, Dir buster scan, and Searchsploit scan. Moreover, the script automatically creates a comprehensive Word document containing all the scan details, allowing testers to initiate the report phase of the web application testing automatically. The script's functionality aims to simplify the web application penetration testing active reconnaissance and information-gathering process, making it faster and more effective for testers.

Overall, this Python script offers a powerful and efficient solution for conducting the active reconnaissance aspects of web application penetration testing in accordance with OWASP guidelines. Its automation capabilities not only save users valuable time and effort but also provide a comprehensive reconnaissance analysis of the web application's security strength.

.

Contents

.

1 INTRODUCTION

1.1 BACKGROUND

Performing all parts of a penetration test one after the other can be time-consuming. These phases being, passive reconnaissance, active reconnaissance/scanning, vulnerability detection exploitation, and report.

Doing these steps manually and one at a time might not be the most efficient way for the tester, but this Python script (written in Python3, the latest version) is intended to optimise the process and save time by automating the active reconnaissance, vulnerability detection and report. The tester can run the script which will do all the active scanning by running the tools Nmap, Nikto, WhatWeb, Dirb and Searchsploit. A text files will be created for each tool after the scan. In the output result of these tools, can be found valuable information such as open ports, services and versions by Nmap, interesting addresses with Dirb, and through the result of Searchsploit exploits of the versions of the services found with Nmap can be retrieved. At the end of the script, a Word document will be generated which will contain all the scanning s from each tool. Some of the scanning outputs are too long to just dump into the Word document, that is where parsing comes into play.

The script also uses threads for multithreading. By creating threads, it will split the execution of each tool so that they are done simultaneously instead of one after the other. This significantly shortens the overall time it takes from when the tester executes the script to when the output is passed to Word.

1.2 AIM

The scripting project has been developed to simplify the active reconnaissance stage of a web application testing for penetration testers. It achieves this objective by simplifying, automating, and optimizing the testing process, in that way minimising the amount of user input required and the time to retrieve the information. With this project, the only input required from the user is the web application IP address, making the testing process more efficient and effective.

1.3 ETHICS CONCERNS

This script executes tools that can retrieve compromising information from a web application. The tools are running active reconnaissance and could be flagged as an attempt of intrusion by the defence systems. This can lead to penalties if used without permission. The web application used in this report is fictitious and for educational purposes only.

2 PROGRAM DEVELOPMENT

2.1 GENERAL CODE

The programming language chosen for this scripting project was Python3. It offers better performance, syntaxis and security improvements, unicode support and new features compared to its other versions, making it the ideal choice for this script development.

Several Python libraries were used to run the script and achieved the goal of optimizing the tester's job. Import OS which allows to access functions and methods related to the operating system like manipulating files and obtaining information. Import Document is a class from the python-docx library allowing the creation and manipulation of Word documents. Import re provides regular expressions. Import threading allows Python to work with multiprocessing, executing multiple tasks simultaneously improving efficiency and performance.

The first action the script did was to create a directory using the date and time when the code was executed by the user as the name of the directory. This was intended to be a guide to the tester to know when the scan was performed in the case of changes on the web application. All the output files and the report are meant to be automatically saved in this directory.

To execute the tools the code had to call functions in which each tool and its parameters were set appropriately. Those which required a URL such as Nikto, Dirb and WhatWeb concatenated the IP which was then used to create a variable containing the URL.

As each tool was running simultaneously on an individual thread, once it was done, it held the result in a variable called result_holder while waiting for all the other threads to finish. Once all the threads were done with their respective tools then the function generate_word_document was called.

2.2 WHATWEB FUNCTION

WhatWeb is a tool that identifies the technologies used by web applications. It analyses HTTP headers, JavaScript files and more to retrieve information about the target, making it the ideal tool to retrieve valuable information from the web application.

To run this tool the script first had to transform the IP into a URL by concatenating http://+ IP. Once the tool was done executing, it saved the output into a file called WhatWeb Scan. The last instruction in the function was to hold the result while it waited for the thread to join.

WhatWeb outputs are written in ANSI (American National Standards Institute) which adds colour to the output which posed a significant challenge to the programmer as the WhatWeb.txt file read the text in ASCII and was not able to translate into plaintext. Different attempts were made in order to translate the text into plaintext but were unsuccessful, nevertheless, when using the command CAT on a Linux terminal to open and read the file it came back in plaintext with its colours. Evidence of this can be found in the discussion section under WhatWeb Result

2.3 NMAP SCAN

Nmap (Network mapping) is a well-known tool that is used against a target, (in this case the IP of the web application) to find information about opened ports, services, service versions, operating systems and vulnerabilities when running the appropriate script with the command.

The Nmap function has 2 variables passed called, IP, dir_name and result_holder. The first thing it did was to create the output file named after the scan. There was no need for concatenation as Nmap works with IP. The parameters passed for this scan are the following:

- -sV: parameter used to determine the service versions.
- -O: parameter used to determine the operating system on which the IP is running from.
- -p-: parameter used to scan all the 65,535 ports on TCP.

Once the scan had been executed, which took a while as it is a big scan looking for opened ports, it saved it into a file called nmap.txt and then held a copy of the output in the variable result_holder. The full result of the Nmap scan can be found in section 3 under the Nmap result discussion.

2.4 SEARCHSPLOIT FUNCTION

The tool Searchsploit as the name implies, it searches for known exploits of a specific version or port presented when running the scan. It uses the well-known exploit-db (Exploit Database) to find the vulnerabilities.

To use this tool appropriately it was first needed to find the versions. That is where the function extract_version came into play. The versions from the Nmap output had to be parsed into a new file called versions.txt. The 2 versions that were being searched were APACHE and PHP. Once this was done, the function extract_version called another function run_searchsploit.

The following function will take in 3 variables, dir_name where it saved its result, apache_version and php_version. It ran the tool Searchsploit using both the Apache and PHP versions and saved the output of both searches into a file sploit.txt.

2.5 NIKTO FUNCTION

Nikto is an exhaustive vulnerability scanner used to find vulnerabilities, misconfigurations, and potential weaknesses in web applications. Either being an outdated software, non-hidden and user-accessible directories, cross-site scripting, or SQL injection. It also generates a kind of report that summarises all its findings.

The function that ran this tool is called run_nikto, which received 3 parameters, IP, dir_name directory and result_holder. First, it had to create the file where it was going to store its result. The parameter used by the tool was "Nikto -h URL", so the IP had to be concatenated and converted into a URL as it was also done with the WhatWeb tool. Once the tool was done scanning, it saved its output into the file and waited for other threads to finish and call join. Lastly, the Nikto tool scan was saved into the Word document.

2.6 DIRB FUNCTION

The following tool on the script was Dirb. Dirb stands for directory buster. As its name suggests, this tool's main function is to find interesting directories not hidden within a web application.

The function first created the text file inside the directory just like the previous scans did, then it concatenates the IP converting it into a URL as it was previously explained in the WhatWeb scan. Once the Dirb command was done executing, it saved its output in its appropriate text file.

The original output of the Dirb output was about 20,000 lines, this was because to find the directories Dirb used a wordlist again the IP to verify if those directories existed and were accessible. The output consisted of all the attempts of the wordlist. Saving the entire output into the Word document would not be efficient as some information might not be able to be seen clearly. To solve this problem and make it more convenient for the tester another function was called to parse the information. This function did not alter the original file, instead, it creates its file.

The parsing function will be explained more in detail in the Word document Function as its goal is to provide a clear output of the Dirb scan that can be added to the Word document report.

2.7 WORD DOCUMENT FUNCTION

To create a Word document the Python library python-docx Document was used. The function took the following parameter that was used to create the document and dump the information of the scan performed previously. This Word document serves as a detailed evaluation report on the web application.

There were several instructions for this, they are the following:

- An empty document was created by calling Document()
- The instruction document.add_heading added the level 1 title, in this case, “Web application Testing Report + IP” to the Word document.

- A for loop instruction was passed over all the output of WhatWeb, Nikto, and Nmap outputs. At each iteration of the loop, a level 2 header would be added to the Word and the content of the files was read and copied into the Word.
- The following instruction was still within the loop, it sanitised the text of the output files using a regular expression to eliminate any character that was not between the 0x20 and 0x7E range as not all outputs were in ASCII. The range of 0x20 and 0x7E are printable characters such as letters, numbers, punctuation, and symbols.
- The clean text was then added to the Word document with the instruction document.add_parragraph in the form of paragraphs and text under its respective title.
- After the loop was done, the next instruction was to add the Dirb title in level 2, which was not within the loop as its output was parsed. Once the title was added, the filtered output that was gained by parsing the original Dirb output was then added to the word under the Dirb title as a paragraph.
- Next was to add the Searchsploit output. This was done by adding a level 2 title called Searchsploit Result. In order to add the text, it will read and encode the Searchsploit result using UTF-8 format to handle special characters and ignore any possible errors.
- It reads the content using regular expressions to remove any characters not within the ASCII range table.
- Lastly, it saves the Word document into the same directory as all the other scans.

2.8 MAIN FUNCTION

The main function is where everything came together. Firstly, it asked the tester to enter the IP of the web application, which is then stored in a variable named IP. Then it created the variable from where it called the create_dir function and passed the IP address for its creation. The following instruction was to create the dictionary named result_holder variable, this is the dictionary the tools used to store their result while waiting.

The next instruction created the threads for each one of the following tools to run their functions, WhatWeb, Nmap, Nikto and Dirb functions. It called the function and added the appropriate parameters. Once the threads were created, they were initiated by calling the threadname.start() i.e. whatweb_thead.start(), this allowed them to run simultaneously. The main function waited for all threads to finish their execution using the `join()` method. This ensured that the program did not proceed until all threads had completed their tasks. Once all threads had finished, the code retrieved the results from the `result_holder` dictionary for each tool. Finally, the code called the last function named `generate_word_document` to generate a Word document containing the assessment results obtained from all the tools. At the end, the tool compiled without any error and will be discussed in the following sections

3 General Discussion

3.1 General Discussion

Here is the result of the script. From the image below it can be seen all the outputs are generated by the script. Each output file will be shown in the subsequent section with a brief description of the content.

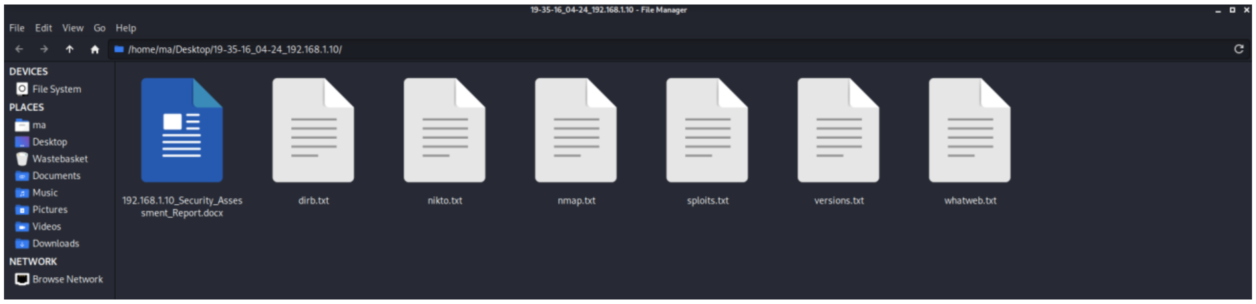
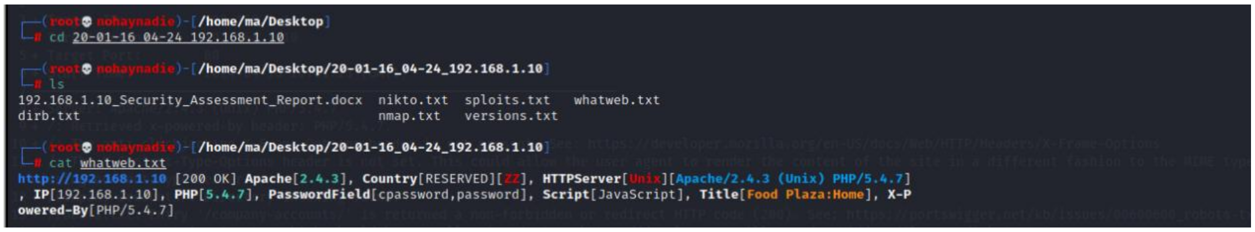


Figure 1: Folder created by the script containing all scans output and Word document report.

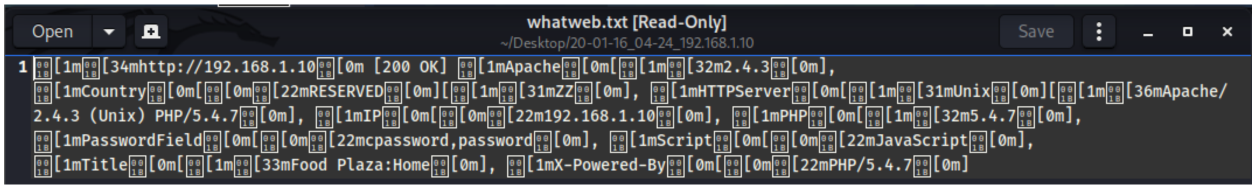
The script aimed to automate and optimise the active reconnaissance of the web application testing and this was successfully done with all the output files and a report-style document being generated within an average time of 39 seconds (the average time scan was run on a Kali Linux VM with 5GB RAM, the time may vary depending on the specification of the device).

3.1.1 WhatWeb Result Discussion

The images below show the content of the WhatWeb scan when using the cat command. As the results are written in ANSI and are mostly made to be viewed from the terminal, it looks neater from the terminal than when opened in a text file.



```
┌──(root💀nohaynadie)-[/home/ma/Desktop]
└─# cd 20-01-16_04-24_192.168.1.10

┌──(root💀nohaynadie)-[/home/ma/Desktop/20-01-16_04-24_192.168.1.10]
└─# ls
192.168.1.10_Security_Assessment_Report.docx  nikto.txt  sploits.txt   whatweb.txt
dirb.txt                                      nmap.txt   versions.txt

┌──(root💀nohaynadie)-[/home/ma/Desktop/20-01-16_04-24_192.168.1.10]
└─# cat whatweb.txt
http://192.168.1.10 [200 OK] Apache[2.4.3], Country[RESERVED][ZZ], HTTPServer[Unix][Apache/2.4.3 (Unix) PHP/5.4.7]
, IP[192.168.1.10], PHP[5.4.7], PasswordField[cpassword,password], Script[JavaScript], Title[Food Plaza:Home], X-P
owered-By[PHP/5.4.7]
```

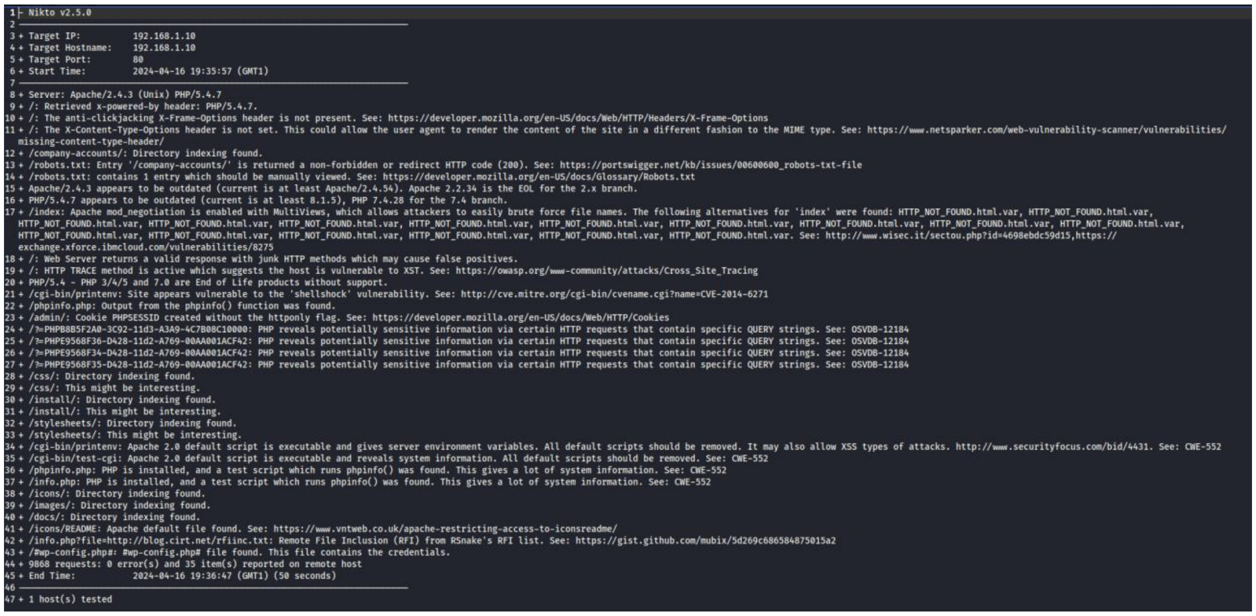


```
␛[1m␛[34mhttp://192.168.1.10␛[0m [200 OK] ␛[1mApache␛[0m[␛[1m␛[32m2.4.3␛[0m],
␛[1mCountry␛[0m[␛[0m␛[22mRESERVED␛[0m][␛[1m␛[31mZZ␛[0m], ␛[1mHTTPServer␛[0m[␛[1m␛[31mUnix␛[0m][␛[1m␛[36mApache/
2.4.3 (Unix) PHP/5.4.7␛[0m], ␛[1mIP␛[0m[␛[0m␛[22m192.168.1.10␛[0m], ␛[1mPHP␛[0m[␛[1m␛[32m5.4.7␛[0m],
␛[1mPasswordField␛[0m[␛[0m␛[22mcpassword,password␛[0m], ␛[1mScript␛[0m[␛[0m␛[22mJavaScript␛[0m],
␛[1mTitle␛[0m[␛[1m␛[33mFood Plaza:Home␛[0m], ␛[1mX-Powered-By␛[0m[␛[0m␛[22mPHP/5.4.7␛[0m]
```

Figures 2 & 3: WhatWeb result viewed from a terminal and viewed from a text file.

3.1.2 Nikto Result Discussion

The Nikto output can be viewed clearly from either terminal or text files as shown in the image below. Information such as directories found, outdated versions, and possible false negatives HTTP responses can be found within the results, showing the tool ran successfully and found various weaknesses and vulnerabilities that need to be reported.

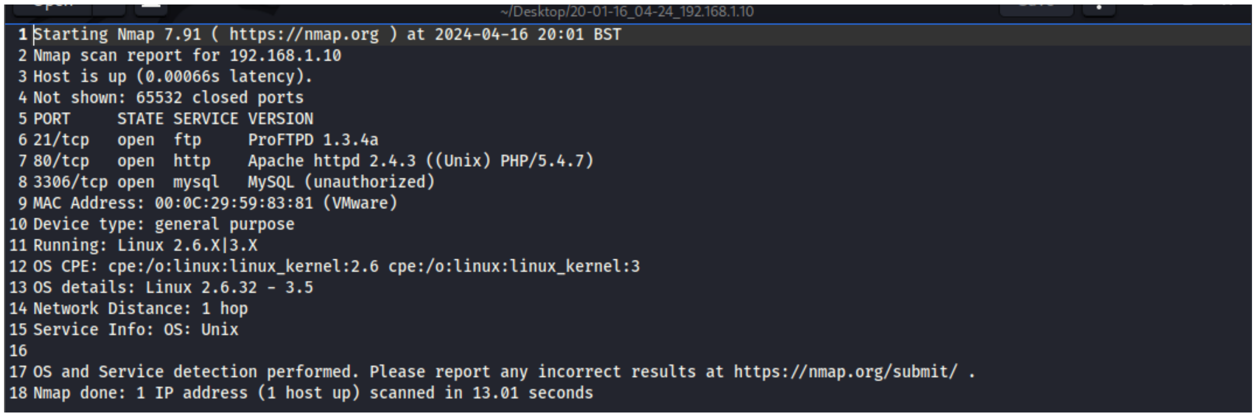


```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.1.10
+ Target Hostname:    192.168.1.10
+ Target Port:        80
+ Start Time:         2024-04-16 19:35:57 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache/2.4.3 (Unix) PHP/5.4.7
+ /: Retrieved x-powered-by header: PHP/5.4.7.
+ /: 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/
+ /company-accounts/: Directory indexing found.
+ /robots.txt: Entry '/company-accounts/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ Apache/2.4.3 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ PHP/5.4.7 appears to be outdated (current is at least 8.1.5), PHP 7.4.28 for the 7.4 branch.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
  HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://
  exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ PHP/5.4 - PHP 3/4/5 and 7.0 are End of Life products without support.
+ /cgi-bin/printenv: Site appears vulnerable to the 'shellshock' vulnerability. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /admin/: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /css/: Directory indexing found.
+ /css/: This might be interesting.
+ /install/: Directory indexing found.
+ /install/: This might be interesting.
+ /stylesheets/: Directory indexing found.
+ /stylesheets/: This might be interesting.
+ /cgi-bin/printenv: Apache 2.0 default script is executable and gives server environment variables. All default scripts should be removed. It may also allow XSS types of attacks. http://www.securityfocus.com/bid/4431. See: CWE-552
+ /cgi-bin/test-cgi: Apache 2.0 default script is executable and reveals system information. All default scripts should be removed. See: CWE-552
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /images/: Directory indexing found.
+ /docs/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /info.php?file=http://blog.cirt.net/rfiinc.txt: Remote File Inclusion (RFI) from RSnake's RFI list. See: https://gist.github.com/mubix/5d269c686584875015a2
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 9868 requests: 0 error(s) and 35 item(s) reported on remote host
+ End Time:           2024-04-16 19:36:47 (GMT1) (50 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 4: Nikto output scan.

3.1.3 Nmap Result Discussion

Here is the result of the Nmap scan when using the following parameter: -O -p- -sV. Information such as, MAC address, operating systems open port, services and versions were found, making this another successful scan by the script.



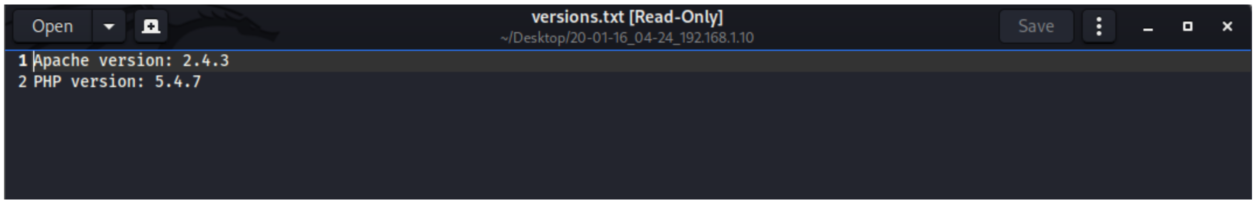
~/Desktop/20-01-16_04-24_192.168.1.10

```
Starting Nmap 7.91 ( https://nmap.org ) at 2024-04-16 20:01 BST
Nmap scan report for 192.168.1.10
Host is up (0.00066s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     ProFTPD 1.3.4a
80/tcp   open  http    Apache httpd 2.4.3 ((Unix) PHP/5.4.7)
3306/tcp open  mysql   MySQL (unauthorized)
MAC Address: 00:0C:29:59:83:81 (VMware)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.5
Network Distance: 1 hop
Service Info: OS: Unix

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.01 seconds
```

Figure 5: Nmap scan result.

3.1.4 Searchsploit Result Discussion

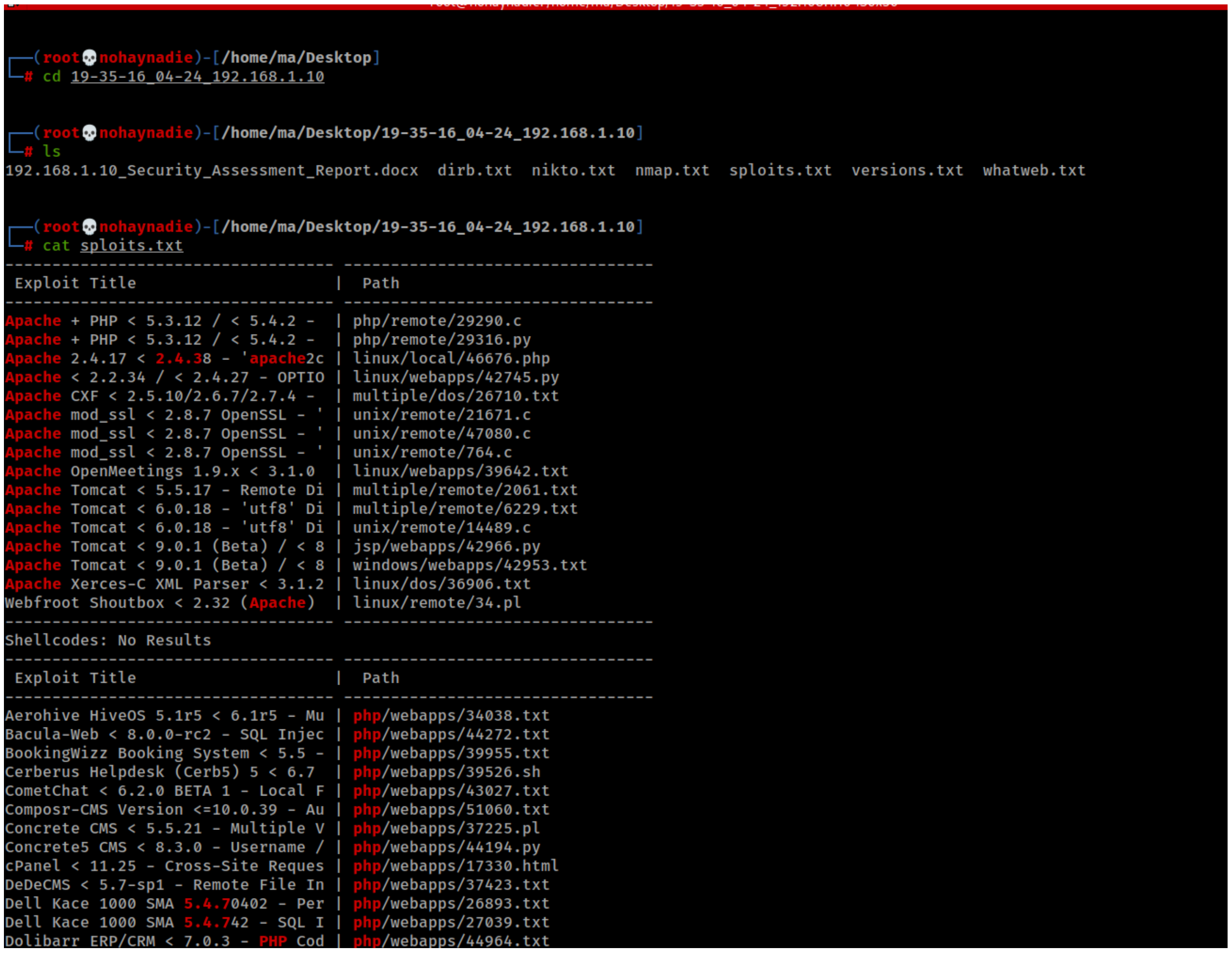
After extracting all the service versions from the Nmap scan output file, they were processed and stored in a separate file named Versions.txt. The creation of the file can be found in Figure 1. Additionally, in the image below it can be observed the actual contents of the file.



```
Apache version: 2.4.3
PHP version: 5.4.7
```

Figure 6: Folder created by the script containing all scans output and Word document report.

The script iterated through the versions above and ran Searchsploit against them. The result of the search can be found below, and the full result can be also found in Figure A2 and Figure A3 located in Appendix A.

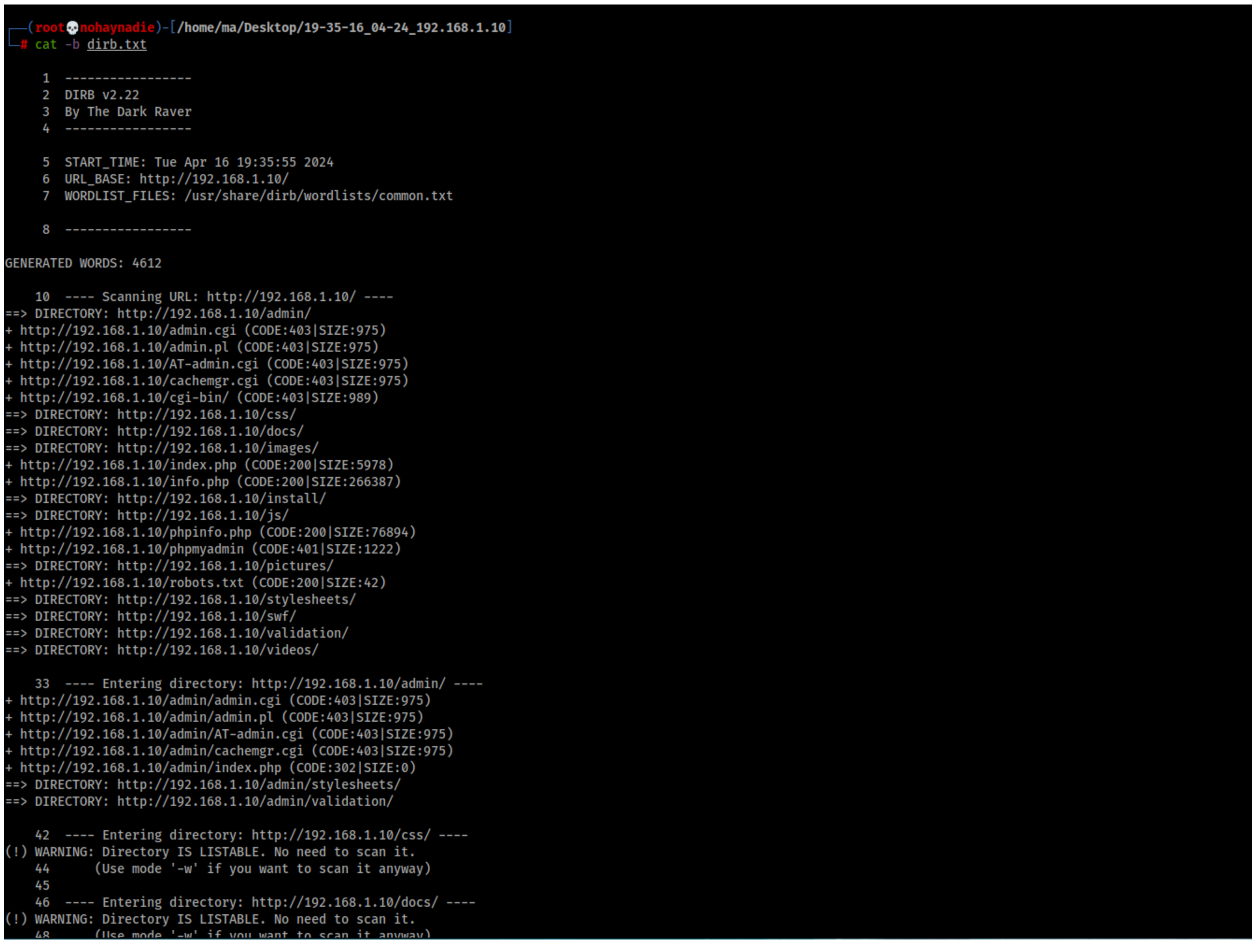


```
┌──(root💀nohaynadie)-[/home/ma/Desktop]
└─# cd 19-35-16_04-24_192.168.1.10

┌──(root💀nohaynadie)-[/home/ma/Desktop/19-35-16_04-24_192.168.1.10]
└─# ls
192.168.1.10_Security_Assessment_Report.docx  dirb.txt  nikto.txt  nmap.txt  sploits.txt  versions.txt  whatweb.txt

┌──(root💀nohaynadie)-[/home/ma/Desktop/19-35-16_04-24_192.168.1.10]
└─# cat sploits.txt
----------------------------------- ---------------------------------
 Exploit Title                     |  Path
----------------------------------- ---------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 -  | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 -  | php/remote/29316.py
Apache 2.4.17 < 2.4.38 - 'apache2c | linux/local/46676.php
Apache < 2.2.34 / < 2.4.27 - OPTIO | linux/webapps/42745.py
Apache CXF < 2.5.10/2.6.7/2.7.4 -  | multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - ' | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - ' | unix/remote/47080.c
Apache mod_ssl < 2.8.7 OpenSSL - ' | unix/remote/764.c
Apache OpenMeetings 1.9.x < 3.1.0  | linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Di | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Di | multiple/remote/6229.txt
Apache Tomcat < 6.0.18 - 'utf8' Di | unix/remote/14489.c
Apache Tomcat < 9.0.1 (Beta) / < 8 | jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8 | windows/webapps/42953.txt
Apache Xerces-C XML Parser < 3.1.2 | linux/dos/36906.txt
Webfroot Shoutbox < 2.32 (Apache)  | linux/remote/34.pl
----------------------------------- ---------------------------------
Shellcodes: No Results
----------------------------------- ---------------------------------
 Exploit Title                     |  Path
----------------------------------- ---------------------------------
Aerohive HiveOS 5.1r5 < 6.1r5 - Mu | php/webapps/34038.txt
Bacula-Web < 8.0.0-rc2 - SQL Injec | php/webapps/44272.txt
BookingWizz Booking System < 5.5 - | php/webapps/39955.txt
Cerberus Helpdesk (Cerb5) 5 < 6.7  | php/webapps/39526.sh
CometChat < 6.2.0 BETA 1 - Local F | php/webapps/43027.txt
Composr-CMS Version <=10.0.39 - Au | php/webapps/51060.txt
Concrete CMS < 5.5.21 - Multiple V | php/webapps/37225.pl
Concrete5 CMS < 8.3.0 - Username / | php/webapps/44194.py
cPanel < 11.25 - Cross-Site Reques | php/webapps/17330.html
DeDeCMS < 5.7-sp1 - Remote File In | php/webapps/37423.txt
Dell Kace 1000 SMA 5.4.70402 - Per | php/webapps/26893.txt
Dell Kace 1000 SMA 5.4.742 - SQL I | php/webapps/27039.txt
Dolibarr ERP/CRM < 7.0.3 - PHP Cod | php/webapps/44964.txt
```

Figure 7: Searchsploit result of both the Apache and the PHP versions.

3.1.5 Dirb Result Discussion

The automated script executed the Dirb command and generated a report, which revealed several directories that were found during the scan. Among the directories, there were common ones such as /doc, /images, /phpinfo.php, /robot.txt and more. If not secured properly, these directories can potentially provide attackers with valuable information and pose a significant security risk.



```
┌──(root💀nohaynadie)-[/home/ma/Desktop/19-35-16_04-24_192.168.1.10]
└─# cat -b dirb.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Apr 16 19:35:55 2024
URL_BASE: http://192.168.1.10/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.10/ ----
==> DIRECTORY: http://192.168.1.10/admin/
+ http://192.168.1.10/admin.cgi (CODE:403|SIZE:975)
+ http://192.168.1.10/admin.pl (CODE:403|SIZE:975)
+ http://192.168.1.10/AT-admin.cgi (CODE:403|SIZE:975)
+ http://192.168.1.10/cachemgr.cgi (CODE:403|SIZE:975)
+ http://192.168.1.10/cgi-bin/ (CODE:403|SIZE:989)
==> DIRECTORY: http://192.168.1.10/css/
==> DIRECTORY: http://192.168.1.10/docs/
==> DIRECTORY: http://192.168.1.10/images/
+ http://192.168.1.10/index.php (CODE:200|SIZE:5978)
+ http://192.168.1.10/info.php (CODE:200|SIZE:266387)
==> DIRECTORY: http://192.168.1.10/install/
==> DIRECTORY: http://192.168.1.10/js/
+ http://192.168.1.10/phpinfo.php (CODE:200|SIZE:76894)
+ http://192.168.1.10/phpmyadmin (CODE:401|SIZE:1222)
==> DIRECTORY: http://192.168.1.10/pictures/
+ http://192.168.1.10/robots.txt (CODE:200|SIZE:42)
==> DIRECTORY: http://192.168.1.10/stylesheets/
==> DIRECTORY: http://192.168.1.10/swf/
==> DIRECTORY: http://192.168.1.10/validation/
==> DIRECTORY: http://192.168.1.10/videos/

---- Entering directory: http://192.168.1.10/admin/ ----
+ http://192.168.1.10/admin/admin.cgi (CODE:403|SIZE:975)
+ http://192.168.1.10/admin/admin.pl (CODE:403|SIZE:975)
+ http://192.168.1.10/admin/AT-admin.cgi (CODE:403|SIZE:975)
+ http://192.168.1.10/admin/cachemgr.cgi (CODE:403|SIZE:975)
+ http://192.168.1.10/admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.1.10/admin/stylesheets/
==> DIRECTORY: http://192.168.1.10/admin/validation/

---- Entering directory: http://192.168.1.10/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.10/docs/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

Figure 8: Dirb scan output.

3.1.6 Word Document Result Discussion

The process of generating the Word document was completed successfully. The document consists of a prominent level 1 headline and includes various subheadings for each tool. It can be observed in Figures A1, A2, and A3 which are located in Appendix A.

3.2 Future Work

This project was developed to work on a Kali Linux running in-built tool. Another version of the script can be done to run various web application testing tools on Windows, providing more flexibility from the platform where a tester can run the script. It would also be interesting to see how the script performs when scanning a bigger or a real life web application.

Multithreading was implemented to optimise the script by creating threads for each tool. Benchmarking the script helps find out the number of threads that make the script most effective. Creating a higher number of threads can degrade performance instead of improving it, which defeats the purpose of multithreading. This depends on factors such as the number of CPU cores, and the nature of the task being performed, among others.

If I had more time and more resources, firstly, I would have refined the output of the WhatWeb tool. As the output is designed to be viewed on ANSI-compliant terminals, it has colours and shapes and as can be seen in Figure 3 and the Word document in Figure B1, the output contains escape codes such as “[1m[34mhttp" instead of the real formatting effects viewed in Figure 2. Furthermore, I would implement other parameters when running the Nmap scan. Exploring the use of parameters such as -vuln to search for more vulnerabilities, scan all UDP ports among others to retrieve more information about the target. More complex choices of tools could have been used. Instead of Dirb automate other tolls such as Dirbuster, ZAP or even Burpsuit. These are some aspects of the code I would have attempted to make my script stronger and more effective for a tester.

Additionally, the script could also be taken a step further, not only limiting the script to active reconnaissance but also running other functions such as opening the web page with selenium or the Python “webbrowser” class to try and enter SQL injection into credentials fields and have the script return whether the login was completed successfully or not, providing the tester information on how the web application handles user input effectively.

Lastly, it could also be possible to try and automate some exploits. Using vulnerabilities found with the Searchsploit tool, those could be passed to Metasploit from where an exploit could be executed in other to gain access to a meterpreter. The same idea can also be applied to Nessus. Further research will be required on how to log into Nessus, start a web application scan, and on how to download the information to the directory with all the outputs from the other tools.

4 REFERENCES

Python file open. Available at: https://www.w3schools.com/python/python_file_handling.asp (Accessed: 1 April 2024).

Python functions. Available at: https://www.w3schools.com/python/python_functions.asp (Accessed: 1 April 2024).

Python 3 advantages • python land tutorial (2022) Python Land. Available at: https://python.land/migrating-from-python-2-to-3/python-3-advantages (Accessed: 1 April 2024).

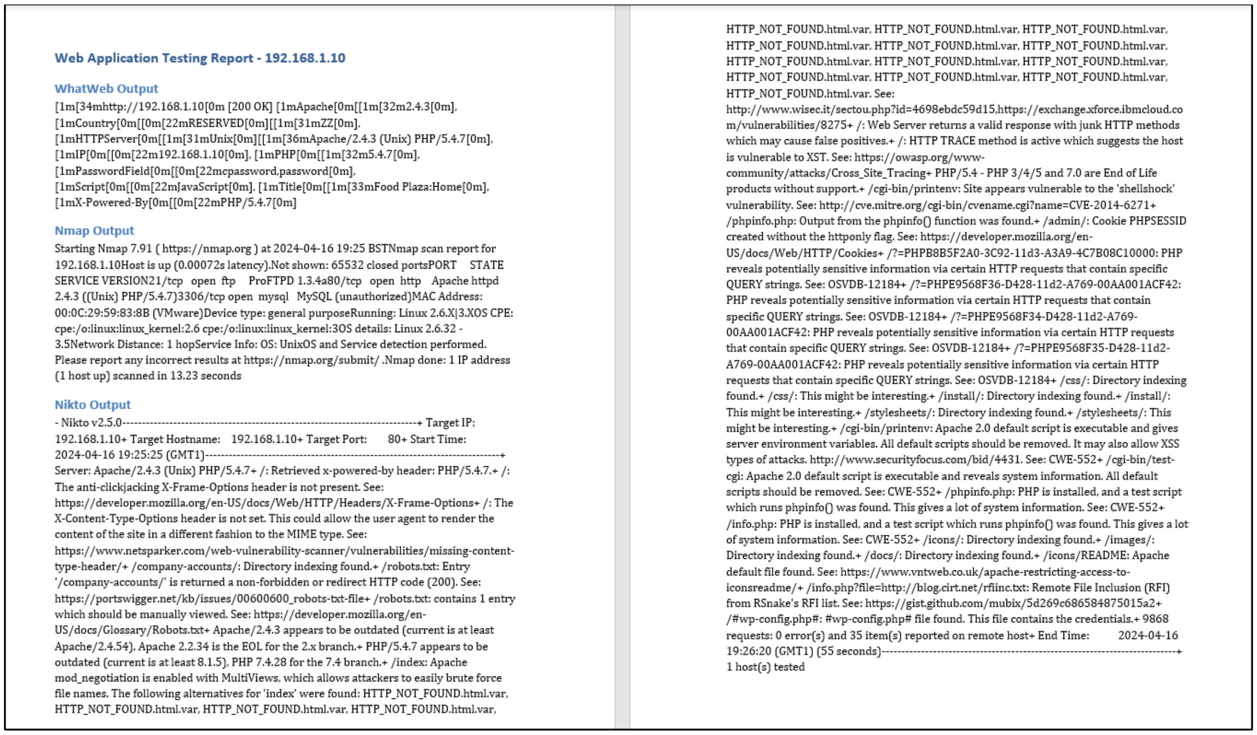
Pitón), por J. (Código (2022) Cómo USAR hilos (threads) en python, Código Pitón. Available at: https://www.codigopiton.com/como-usar-hilos-o-threads-en-python/ (Accessed: 10 April 2024).

Threading - paralelismo Basado en hilos¶ (no date) threading - Paralelismo basado en hilos - documentación de Python - 3.8.18. Available at: https://docs.python.org/es/3.8/library/threading.html (Accessed: 10 April 2024).

GeeksforGeeks (2021) Python script to open a web browser, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/python-script-to-open-a-web-browser/ (Accessed: 14 April 2024).

5 APPENDICES

Appendix A - Word document report



Web Application Testing Report - 192.168.1.10

WhatWeb Output

```
[1m[34mhttp://192.168.1.10[0m [200 OK] [1mApache[0m[[1m[32m2.4.3[0m],
[1mCountry[0m[[0m[22mRESERVED[0m][[1m[31mZZ[0m],
[1mHTTPServer[0m[[1m[31mUnix[0m][[1m[36mApache/2.4.3 (Unix) PHP/5.4.7[0m],
[1mIP[0m[[0m[22m192.168.1.10[0m], [1mPHP[0m[[1m[32m5.4.7[0m],
[1mPasswordField[0m[[0m[22mcpassword,password[0m],
[1mScript[0m[[0m[22mJavaScript[0m], [1mTitle[0m[[1m[33mFood Plaza:Home[0m],
[1mX-Powered-By[0m[[0m[22mPHP/5.4.7[0m]
```

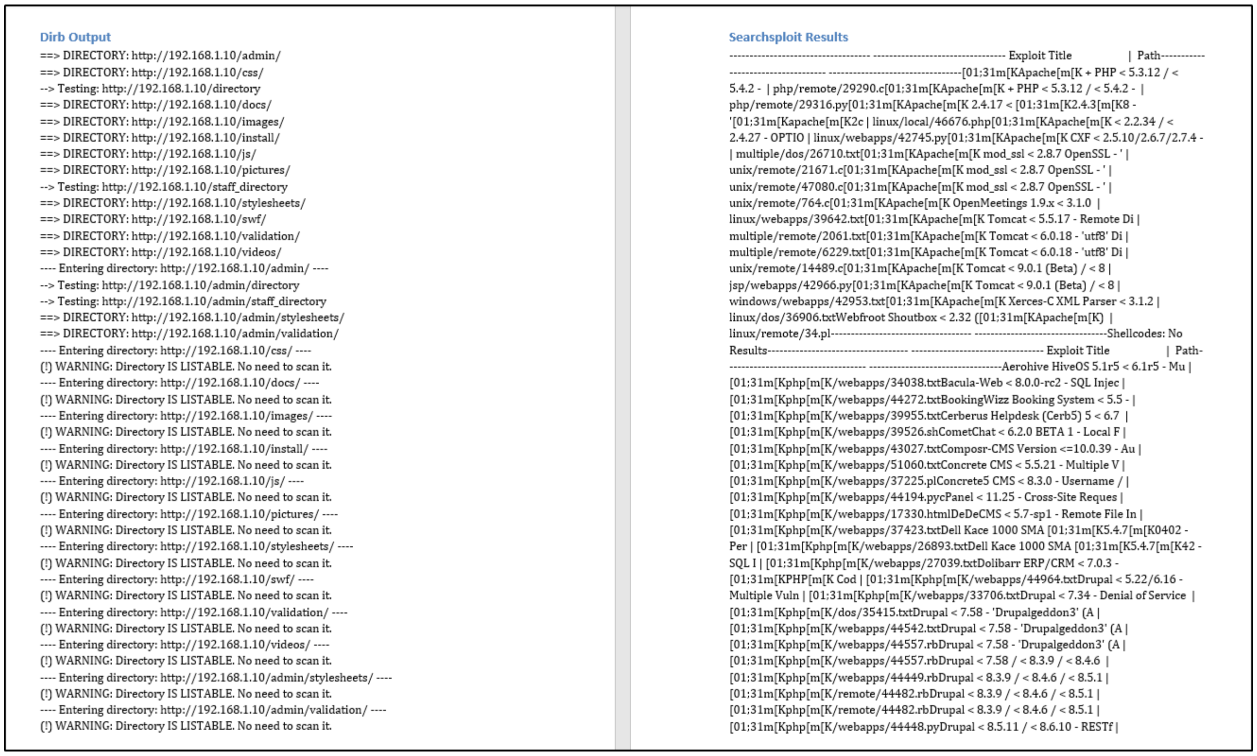
Nmap Output

```
Starting Nmap 7.91 ( https://nmap.org ) at 2024-04-16 19:25 BSTNmap scan report for
192.168.1.10Host is up (0.00072s latency).Not shown: 65532 closed portsPORT     STATE
SERVICE VERSION21/tcp   open  ftp    ProFTPD 1.3.4a80/tcp   open  http    Apache httpd
2.4.3 ((Unix) PHP/5.4.7)3306/tcp open  mysql   MySQL (unauthorized)MAC Address:
00:0C:29:59:83:8B (VMware)Device type: general purposeRunning: Linux 2.6.X|3.XOS CPE:
cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3OS details: Linux 2.6.32 -
3.5Network Distance: 1 hopService Info: OS: UnixOS and Service detection performed.
Please report any incorrect results at https://nmap.org/submit/ .Nmap done: 1 IP address
(1 host up) scanned in 13.23 seconds
```

Nikto Output

```
- Nikto v2.5.0---------------------------------------------------------------------------+ Target IP:
192.168.1.10+ Target Hostname:    192.168.1.10+ Target Port:       80+ Start Time:
2024-04-16 19:25:25 (GMT1)---------------------------------------------------------------------------+
Server: Apache/2.4.3 (Unix) PHP/5.4.7+ /: Retrieved x-powered-by header: PHP/5.4.7.+ /:
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/+ /company-accounts/: Directory indexing found.+ /robots.txt: Entry
'/company-accounts/' is returned a non-forbidden or redirect HTTP code (200). See:
https://portswigger.net/kb/issues/00600600_robots-txt-file+ /robots.txt: contains 1 entry
which should be manually viewed. See: https://developer.mozilla.org/en-
US/docs/Glossary/Robots.txt+ Apache/2.4.3 appears to be outdated (current is at least
Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.+ PHP/5.4.7 appears to be
outdated (current is at least 8.1.5), PHP 7.4.28 for the 7.4 branch.+ /index: Apache
mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force
file names. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var. See:
http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.co
m/vulnerabilities/8275+ /: Web Server returns a valid response with junk HTTP methods
which may cause false positives.+ /: HTTP TRACE method is active which suggests the host
is vulnerable to XST. See: https://owasp.org/www-
community/attacks/Cross_Site_Tracing+ PHP/5.4 - PHP 3/4/5 and 7.0 are End of Life
products without support.+ /cgi-bin/printenv: Site appears vulnerable to the 'shellshock'
vulnerability. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271+
/phpinfo.php: Output from the phpinfo() function was found.+ /admin/: Cookie PHPSESSID
created without the httponly flag. See: https://developer.mozilla.org/en-
US/docs/Web/HTTP/Cookies+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP
reveals potentially sensitive information via certain HTTP requests that contain specific
QUERY strings. See: OSVDB-12184+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42:
PHP reveals potentially sensitive information via certain HTTP requests that contain
specific QUERY strings. See: OSVDB-12184+ /?=PHPE9568F34-D428-11d2-A769-
00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests
that contain specific QUERY strings. See: OSVDB-12184+ /?=PHPE9568F35-D428-11d2-
A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP
requests that contain specific QUERY strings. See: OSVDB-12184+ /css/: Directory indexing
found.+ /css/: This might be interesting.+ /install/: Directory indexing found.+ /install/:
This might be interesting.+ /stylesheets/: Directory indexing found.+ /stylesheets/: This
might be interesting.+ /cgi-bin/printenv: Apache 2.0 default script is executable and gives
server environment variables. All default scripts should be removed. It may also allow XSS
types of attacks. http://www.securityfocus.com/bid/4431. See: CWE-552+ /cgi-bin/test-
cgi: Apache 2.0 default script is executable and reveals system information. All default
scripts should be removed. See: CWE-552+ /phpinfo.php: PHP is installed, and a test script
which runs phpinfo() was found. This gives a lot of system information. See: CWE-552+
/info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot
of system information. See: CWE-552+ /icons/: Directory indexing found.+ /images/:
Directory indexing found.+ /docs/: Directory indexing found.+ /icons/README: Apache
default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-
iconsreadme/+ /info.php?file=http://blog.cirt.net/rfiinc.txt: Remote File Inclusion (RFI)
from RSnake's RFI list. See: https://gist.github.com/mubix/5d269c686584875015a2+
/#wp-config.php#: #wp-config.php# file found. This file contains the credentials.+ 9868
requests: 0 error(s) and 35 item(s) reported on remote host+ End Time:         2024-04-16
19:26:20 (GMT1) (55 seconds)---------------------------------------------------------------------------+
1 host(s) tested
```

Figure A1: WhatWeb, Nmap, and Nikto output exported into the Word document.



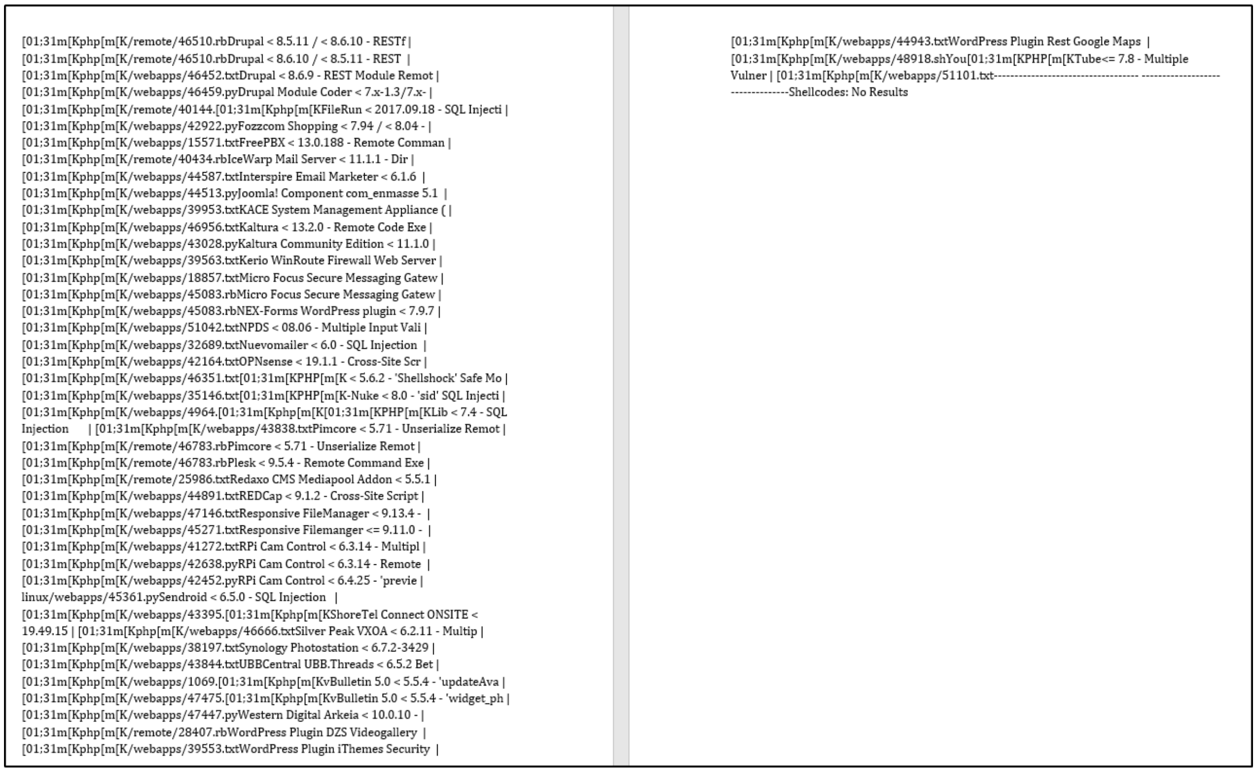
Dirb Output

```
==> DIRECTORY: http://192.168.1.10/admin/
==> DIRECTORY: http://192.168.1.10/css/
--> Testing: http://192.168.1.10/directory
==> DIRECTORY: http://192.168.1.10/docs/
==> DIRECTORY: http://192.168.1.10/images/
==> DIRECTORY: http://192.168.1.10/install/
==> DIRECTORY: http://192.168.1.10/js/
==> DIRECTORY: http://192.168.1.10/pictures/
--> Testing: http://192.168.1.10/staff_directory
==> DIRECTORY: http://192.168.1.10/stylesheets/
==> DIRECTORY: http://192.168.1.10/swf/
==> DIRECTORY: http://192.168.1.10/validation/
==> DIRECTORY: http://192.168.1.10/videos/
---- Entering directory: http://192.168.1.10/admin/ ----
--> Testing: http://192.168.1.10/admin/directory
--> Testing: http://192.168.1.10/admin/staff_directory
==> DIRECTORY: http://192.168.1.10/admin/stylesheets/
==> DIRECTORY: http://192.168.1.10/admin/validation/
---- Entering directory: http://192.168.1.10/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/docs/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/install/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/pictures/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/stylesheets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/swf/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/validation/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/videos/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/admin/stylesheets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
---- Entering directory: http://192.168.1.10/admin/validation/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

Searchsploit Results

```
---------------------------------- -------------------------------- Exploit Title                | Path-----------
------------------------ --------------------------------[01;31m[KApache[m[K + PHP < 5.3.12 / <
5.4.2 -  | php/remote/29290.c[01;31m[KApache[m[K + PHP < 5.3.12 / < 5.4.2 -  |
php/remote/29316.py[01;31m[KApache[m[K 2.4.17 < [01;31m[K2.4.3[m[K8 -
'[01;31m[Kapache[m[K2c | linux/local/46676.php[01;31m[KApache[m[K < 2.2.34 / <
2.4.27 - OPTIO | linux/webapps/42745.py[01;31m[KApache[m[K CXF < 2.5.10/2.6.7/2.7.4 -
| multiple/dos/26710.txt[01;31m[KApache[m[K mod_ssl < 2.8.7 OpenSSL - ' |
unix/remote/21671.c[01;31m[KApache[m[K mod_ssl < 2.8.7 OpenSSL - ' |
unix/remote/47080.c[01;31m[KApache[m[K mod_ssl < 2.8.7 OpenSSL - ' |
unix/remote/764.c[01;31m[KApache[m[K OpenMeetings 1.9.x < 3.1.0  |
linux/webapps/39642.txt[01;31m[KApache[m[K Tomcat < 5.5.17 - Remote Di |
multiple/remote/2061.txt[01;31m[KApache[m[K Tomcat < 6.0.18 - 'utf8' Di |
multiple/remote/6229.txt[01;31m[KApache[m[K Tomcat < 6.0.18 - 'utf8' Di |
unix/remote/14489.c[01;31m[KApache[m[K Tomcat < 9.0.1 (Beta) / < 8 |
jsp/webapps/42966.py[01;31m[KApache[m[K Tomcat < 9.0.1 (Beta) / < 8 |
windows/webapps/42953.txt[01;31m[KApache[m[K Xerces-C XML Parser < 3.1.2 |
linux/dos/36906.txtWebfroot Shoutbox < 2.32 ([01;31m[KApache[m[K)  |
linux/remote/34.pl---------------------------------- --------------------------------Shellcodes: No
Results---------------------------------- -------------------------------- Exploit Title                |  Path-
---------------------------------- --------------------------------Aerohive HiveOS 5.1r5 < 6.1r5 - Mu |
[01;31m[Kphp[m[K/webapps/34038.txtBacula-Web < 8.0.0-rc2 - SQL Injec |
[01;31m[Kphp[m[K/webapps/44272.txtBookingWizz Booking System < 5.5 - |
[01;31m[Kphp[m[K/webapps/39955.txtCerberus Helpdesk (Cerb5) 5 < 6.7  |
[01;31m[Kphp[m[K/webapps/39526.shCometChat < 6.2.0 BETA 1 - Local F |
[01;31m[Kphp[m[K/webapps/43027.txtComposr-CMS Version <=10.0.39 - Au |
[01;31m[Kphp[m[K/webapps/51060.txtConcrete CMS < 5.5.21 - Multiple V |
[01;31m[Kphp[m[K/webapps/37225.plConcrete5 CMS < 8.3.0 - Username / |
[01;31m[Kphp[m[K/webapps/44194.pycPanel < 11.25 - Cross-Site Reques |
[01;31m[Kphp[m[K/webapps/17330.htmlDeDeCMS < 5.7-sp1 - Remote File In |
[01;31m[Kphp[m[K/webapps/37423.txtDell Kace 1000 SMA [01;31m[K5.4.7[m[K0402 -
Per | [01;31m[Kphp[m[K/webapps/26893.txtDell Kace 1000 SMA [01;31m[K5.4.7[m[K42 -
SQL I | [01;31m[Kphp[m[K/webapps/27039.txtDolibarr ERP/CRM < 7.0.3 -
[01;31m[KPHP[m[K Cod | [01;31m[Kphp[m[K/webapps/44964.txtDrupal < 5.22/6.16 -
Multiple Vuln | [01;31m[Kphp[m[K/webapps/33706.txtDrupal < 7.34 - Denial of Service  |
[01;31m[Kphp[m[K/dos/35415.txtDrupal < 7.58 - 'Drupalgeddon3' (A |
[01;31m[Kphp[m[K/webapps/44542.txtDrupal < 7.58 - 'Drupalgeddon3' (A |
[01;31m[Kphp[m[K/webapps/44557.rbDrupal < 7.58 - 'Drupalgeddon3' (A |
[01;31m[Kphp[m[K/webapps/44557.rbDrupal < 7.58 / < 8.3.9 / < 8.4.6  |
[01;31m[Kphp[m[K/webapps/44449.rbDrupal < 8.3.9 / < 8.4.6 / < 8.5.1 |
[01;31m[Kphp[m[K/remote/44482.rbDrupal < 8.3.9 / < 8.4.6 / < 8.5.1 |
[01;31m[Kphp[m[K/remote/44482.rbDrupal < 8.3.9 / < 8.4.6 / < 8.5.1 |
[01;31m[Kphp[m[K/webapps/44448.pyDrupal < 8.5.11 / < 8.6.10 - RESTf |
```

Figure A2: Dirb and Searchsploit output exported into the Word document.



```
[01;31m[Kphp[m[K/remote/46510.rbDrupal < 8.5.11 / < 8.6.10 - RESTf |
[01;31m[Kphp[m[K/remote/46510.rbDrupal < 8.6.10 / < 8.5.11 - REST  |
[01;31m[Kphp[m[K/webapps/46452.txtDrupal < 8.6.9 - REST Module Remot |
[01;31m[Kphp[m[K/webapps/46459.pyDrupal Module Coder < 7.x-1.3/7.x- |
[01;31m[Kphp[m[K/remote/40144.[01;31m[Kphp[m[KFileRun < 2017.09.18 - SQL Injecti |
[01;31m[Kphp[m[K/webapps/42922.pyFozzcom Shopping < 7.94 / < 8.04 - |
[01;31m[Kphp[m[K/webapps/15571.txtFreePBX < 13.0.188 - Remote Comman |
[01;31m[Kphp[m[K/remote/40434.rbIceWarp Mail Server < 11.1.1 - Dir |
[01;31m[Kphp[m[K/webapps/44587.txtInterspire Email Marketer < 6.1.6  |
[01;31m[Kphp[m[K/webapps/44513.pyJoomla! Component com_enmasse 5.1  |
[01;31m[Kphp[m[K/webapps/39953.txtKACE System Management Appliance ( |
[01;31m[Kphp[m[K/webapps/46956.txtKaltura < 13.2.0 - Remote Code Exe |
[01;31m[Kphp[m[K/webapps/43028.pyKaltura Community Edition < 11.1.0 |
[01;31m[Kphp[m[K/webapps/39563.txtKerio WinRoute Firewall Web Server |
[01;31m[Kphp[m[K/webapps/18857.txtMicro Focus Secure Messaging Gatew |
[01;31m[Kphp[m[K/webapps/45083.rbMicro Focus Secure Messaging Gatew |
[01;31m[Kphp[m[K/webapps/45083.rbNEX-Forms WordPress plugin < 7.9.7 |
[01;31m[Kphp[m[K/webapps/51042.txtNPDS < 08.06 - Multiple Input Vali |
[01;31m[Kphp[m[K/webapps/32689.txtNuevomailer < 6.0 - SQL Injection  |
[01;31m[Kphp[m[K/webapps/42164.txtOPNsense < 19.1.1 - Cross-Site Scr |
[01;31m[Kphp[m[K/webapps/46351.txt[01;31m[KPHP[m[K < 5.6.2 - 'Shellshock' Safe Mo |
[01;31m[Kphp[m[K/webapps/35146.txt[01;31m[KPHP[m[K-Nuke < 8.0 - 'sid' SQL Injecti |
[01;31m[Kphp[m[K/webapps/4964.[01;31m[Kphp[m[K[01;31m[KPHP[m[KLib < 7.4 - SQL
Injection      | [01;31m[Kphp[m[K/webapps/43838.txtPimcore < 5.71 - Unserialize Remot |
[01;31m[Kphp[m[K/remote/46783.rbPimcore < 5.71 - Unserialize Remot |
[01;31m[Kphp[m[K/remote/46783.rbPlesk < 9.5.4 - Remote Command Exe |
[01;31m[Kphp[m[K/remote/25986.txtRedaxo CMS Mediapool Addon < 5.5.1 |
[01;31m[Kphp[m[K/webapps/44891.txtREDCap < 9.1.2 - Cross-Site Script |
[01;31m[Kphp[m[K/webapps/47146.txtResponsive FileManager < 9.13.4 -  |
[01;31m[Kphp[m[K/webapps/45271.txtResponsive Filemanger <= 9.11.0 -  |
[01;31m[Kphp[m[K/webapps/41272.txtRPi Cam Control < 6.3.14 - Multipl |
[01;31m[Kphp[m[K/webapps/42638.pyRPi Cam Control < 6.3.14 - Remote  |
[01;31m[Kphp[m[K/webapps/42452.pyRPi Cam Control < 6.4.25 - 'previe |
linux/webapps/45361.pySendroid < 6.5.0 - SQL Injection   |
[01;31m[Kphp[m[K/webapps/43395.[01;31m[Kphp[m[KShoreTel Connect ONSITE <
19.49.15 | [01;31m[Kphp[m[K/webapps/46666.txtSilver Peak VXOA < 6.2.11 - Multip |
[01;31m[Kphp[m[K/webapps/38197.txtSynology Photostation < 6.7.2-3429 |
[01;31m[Kphp[m[K/webapps/43844.txtUBBCentral UBB.Threads < 6.5.2 Bet |
[01;31m[Kphp[m[K/webapps/1069.[01;31m[Kphp[m[KvBulletin 5.0 < 5.5.4 - 'updateAva |
[01;31m[Kphp[m[K/webapps/47475.[01;31m[Kphp[m[KvBulletin 5.0 < 5.5.4 - 'widget_ph |
[01;31m[Kphp[m[K/webapps/47447.pyWestern Digital Arkeia < 10.0.10 - |
[01;31m[Kphp[m[K/remote/28407.rbWordPress Plugin DZS Videogallery  |
[01;31m[Kphp[m[K/webapps/39553.txtWordPress Plugin iThemes Security  |
```

```
[01;31m[Kphp[m[K/webapps/44943.txtWordPress Plugin Rest Google Maps  |
[01;31m[Kphp[m[K/webapps/48918.shYou[01;31m[KPHP[m[KTube<= 7.8 - Multiple
Vulner | [01;31m[Kphp[m[K/webapps/51101.txt---------------------------------- ------------------
--------------Shellcodes: No Results
```

Figure A3: More Searchsploit output exported into the Word document.

APPENDIX B – FULL CODE

```python
import os  #operating system library
from docx import Document  #Word documents library
from datetime import datetime
import re  #regular expressions library
import threading  #multi-threading

def create_dir(ip): # create main direcoty with ip, date and time
    current_datetime = datetime.now().strftime("%H-%M-%d_%m-%y")#Get the current date and time
    dir_name = f"{current_datetime}_{ip}"     # Combine date and ip
    os.makedirs(dir_name)      # Create the directory
    return dir_name

def run_whatweb(ip, dir_name, result_holder): # Function to execute WhatWeb and save output
    output_file = os.path.join(dir_name, "whatweb.txt") # Define the path for the output file
    url = "http://" + ip  # concatenate and turn ip to url
    os.system(f"whatweb {url} > {output_file}")  # run command and save the output
    print(f"WhatWeb output was saved to {output_file}")
    result_holder['whatweb'] = output_file

# same comments from run_whatweb are applied here
def run_nmap(ip, dir_name, result_holder):
    output_file = os.path.join(dir_name, "nmap.txt")
    os.system(f"nmap -sV -O -p- {ip} > {output_file}")
    print(f"Nmap output was saved to {output_file}")
    result_holder['nmap'] = output_file

    # Call the function extract and write versions from the Nmap output
    with open(output_file, 'r') as f:
        nmap_output = f.read()
        extract_versions(dir_name, nmap_output)

# function to extract versions from Nmap output
def extract_versions(dir_name, nmap_output):
    # Regular expressions to extract Apache and PHP versions
    apache_regex = r"Apache httpd ([\d.]+)"
    php_regex = r"PHP/([\d.]+)"

    # wxtract apache and PHP versions using regular expressions
    apache_match = re.search(apache_regex, nmap_output)
    apache_version = apache_match.group(1) if apache_match else ""

    php_match = re.search(php_regex, nmap_output)
    php_version = php_match.group(1) if php_match else ""

    # write Apache and PHP versions to a file
    with open(os.path.join(dir_name, "versions.txt"), "w") as f:
        f.write(f"Apache version: {apache_version}\n")
        f.write(f"PHP version: {php_version}\n")
    print("Versions extracted to 'versions.txt'.")

    # run searchsploit for Apache and PHP versions
    run_searchsploit(dir_name, apache_version, php_version)

def run_searchsploit(dir_name, apache_version, php_version):
    # execute searchsploit for apache and PHP versions and save the results to a file
    os.system(f"searchsploit     'Apache     {apache_version}'     >     {os.path.join(dir_name,
'sploits.txt')}")
    os.system(f"searchsploit 'PHP {php_version}' >> {os.path.join(dir_name, 'sploits.txt')}")
    sploits_file = os.path.join(dir_name, "sploits.txt")
    print("Searchsploit results were saved to 'sploits.txt'.")
    return '\n'.join(sploits_file)

# same comments from run_whatweb are applied here
def run_nikto(ip, dir_name, result_holder):
    output_file = os.path.join(dir_name, "nikto.txt")
    url = "http://" + ip
    os.system(f"nikto -h {url} > {output_file}")
    print(f"Nikto output was saved to {output_file}")
    result_holder['nikto'] = output_file

# same comments from run_whatweb are applied here
def run_dirb(ip, dir_name, result_holder):
    output_file = os.path.join(dir_name, "dirb.txt")
    url = "http://" + ip
    os.system(f"dirb {url} > {output_file}")
    print(f"Dirb output was saved to {output_file}")
    result_holder['dirb'] = output_file

# function to parse Dirb output and filter relevant lines
def parse_dirb_output(output_file):
    try:
        # Open the Dirb output file
        with open(output_file, 'r') as f:
            # Read all lines from the file
            lines = f.readlines()
            # Filter lines containing the word 'directory' and remove whitespaces
            filtered_lines = [line.strip() for line in lines if 'directory' in line.lower()]
        # turn filtered lines into a single string separated by newline characters
        return '\n'.join(filtered_lines)
    except Exception as e:
        print(f"Error parsing Dirb output: {e}")
        return ''

#generate a Word document report with the results
def generate_word_document(ip, dir_name, whatweb_output, nmap_output, nikto_output, dirb_output):
    document = Document() # create a new Word document
    document.add_heading(f"Web Application Testing Report - {ip}", level=1) # add a heading

    # loop through each tool output and add it to the document
    for name, output_file in [("WhatWeb Output", whatweb_output), ("Nmap Output", nmap_output),
("Nikto Output", nikto_output)]:
        document.add_heading(name, level=2)# add a subheading with the tool name
        with open(output_file, "r", encoding="utf-8", errors="ignore") as f: # open the output
file
            text = re.sub(r'[^\x20-\x7E]', '', f.read())   # read the file contents and remove
non-ASCII characters
            document.add_paragraph(text) # add the cleaned text to the document as a paragraph

    document.add_heading("Dirb Output", level=2) #add a subheading with the tool name
    dirb_filtered_output = parse_dirb_output(dirb_output) # Parse and filter Dirb output
    document.add_paragraph(dirb_filtered_output) # Add the filtered Dirb output to the document
as a paragraph

    document.add_heading("Searchsploit Results", level=2) # create sploits output file
    sploits_file_path = os.path.join(dir_name, "sploits.txt")
    with open(sploits_file_path, "r", encoding="utf-8", errors="ignore") as f:
        sploits_text = re.sub(r'[^\x20-\x7E]', '', f.read()) # read the contents and remove non-
ASCII characters
        document.add_paragraph(sploits_text)  # add the cleaned text to the document as a
paragraph


        # Define the path for the final Word document
        output_file = os.path.join(dir_name, f"{ip}_Security_Assessment_Report.docx")
        document.save(output_file) # Save the document
    print(f"Word document saved as {output_file}")


def main():
    # ask for ip
    ip = input("Introcude the target IP: ")
    # Create a directory to store assessment results
    dir_name = create_dir(ip)

    # Create a dictionary to store the results of the tools
    result_holder = {}

    # Create threads to execute each tool simultaneously
    whatweb_thread = threading.Thread(target=run_whatweb, args=(ip, dir_name, result_holder))
    nmap_thread = threading.Thread(target=run_nmap, args=(ip, dir_name, result_holder))
    nikto_thread = threading.Thread(target=run_nikto, args=(ip, dir_name, result_holder))
    dirb_thread = threading.Thread(target=run_dirb, args=(ip, dir_name, result_holder))

    # Start the threads
    whatweb_thread.start()
    nmap_thread.start()
    nikto_thread.start()
    dirb_thread.start()

    # Wait for all threads to finish
    whatweb_thread.join()
    nmap_thread.join()
    nikto_thread.join()
    dirb_thread.join()

    # Extract the results from the dictionary
    whatweb_output = result_holder.get('whatweb', '')
    nmap_output = result_holder.get('nmap', '')
    nikto_output = result_holder.get('nikto', '')
    dirb_output = result_holder.get('dirb', '')

    # Generate the Word document with the results
    generate_word_document(ip, dir_name, whatweb_output, nmap_output, nikto_output, dirb_output)

if __name__ == "__main__":
    main()
```