

A Comprehensive Security Audit On

ON

<https://www.jjmcoe.ac.in/>

Carried out at



**CDAC Software Training and Development Centre
Thiruvananthapuram**

Under the Guidance of

Jayaram P Sir

Senior Cyber Security Specialist at CDAC

Thiruvananthapuram, Kerala, India

Submitted by

Pushkar Zend (240860940032)

Ganesh Shelke (240860940017)

Mayur Pawar (240860940029)

Om Chaudhari(240860940026)

Prathamesh Ingale(240860940031)

Shreyes Pol(240860940046)

CONTENTS

1. Abstract	3
2. Introduction	4
3. Literature Survey	5
4. Scope and Objectives	6
5. Methodology	7
6. Phases of Security Audit	8
7. Reconnaissance	9
8. NMAP	12
9. Nikto	15
10. Vulnerabilities, Description, POC and Recommendations	20
11. Conclusion	33
12. References	34

ABSTRACT

A security audit is a systematic examination of an organization's information systems, technology infrastructure, processes, and policies to evaluate the effectiveness of its security measures, identify vulnerabilities, and ensure compliance with industry best practices and standards. This project focuses on conducting a comprehensive penetration test of the web application belonging to the engineering college at <https://www.jjmcoe.ac.in/>. The primary objective is to identify vulnerabilities within the application and provide actionable recommendations to enhance its overall security posture.

The project involves DNS reconnaissance and detailed mapping of the target application, followed by vulnerability assessment using both manual testing—guided by the OWASP Top 10 framework—and automated scanning tools such as Nessus and others. By employing more than eight well-known tools, we aim to ensure comprehensive coverage and minimize the risk of overlooking critical vulnerabilities.

The scope of this project includes assessing the web application's security posture, identifying potential attack vectors, analyzing the impact of discovered vulnerabilities, and developing a detailed remediation plan. The outcome will include a set of actionable recommendations to safeguard the application's sensitive data and ensure its secure operation.

Keywords: Security Auditing, Web Application Security, Vulnerabilities, DNS Reconnaissance, OWASP Top 10.

INTRODUCTION

In an era of rapid technological advancements and digital transformation, ensuring the security and integrity of systems, networks, and data is more critical than ever. As organizations, including educational institutions, increasingly rely on digital infrastructure for operations, communication, and the storage of sensitive information, the risks and vulnerabilities associated with such reliance continue to grow. Adopting a proactive approach to identifying, mitigating, and managing these risks is vital for protecting an organization's assets, reputation, and stakeholder trust.

This **Security Audit Project Report** provides a detailed evaluation of the security posture of the digital ecosystem supporting the <https://www.jjmcoe.ac.in/> platform. The primary objective of this security audit was to systematically assess the effectiveness of existing security measures, policies, and practices, while offering actionable recommendations that align with industry standards and best practices. Through a comprehensive analysis of the web application's technology stack, data handling mechanisms, and access control protocols, the audit seeks to deliver valuable insights for strengthening the platform's overall security framework.

The report is structured to present the scope of the audit, the methodologies utilized, key findings, and a set of remediation strategies. Furthermore, it highlights the importance of fostering a security-centric culture within the organization and emphasizes the need for continuous monitoring and adaptation to counter the ever-evolving cyber threat landscape.

In the sections that follow, we will delve into the specifics of the security audit, detailing its relevance in addressing modern cyber threats. We will also outline the collaborative efforts of the audit team in safeguarding the confidentiality, integrity, and availability of the <https://www.jjmcoe.ac.in/> platform's critical assets.

LITERATURE SURVEY

The OWASP Top 10 is a well-known list of the top 10 most critical security risks commonly found in web applications. Including these in your Security Audit Project Report helps to highlight key vulnerabilities that should be addressed. As of my last update in September 2021, here's the OWASP Top 10 list:

OWASP Top 10 Security Risks - 2021

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failure
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

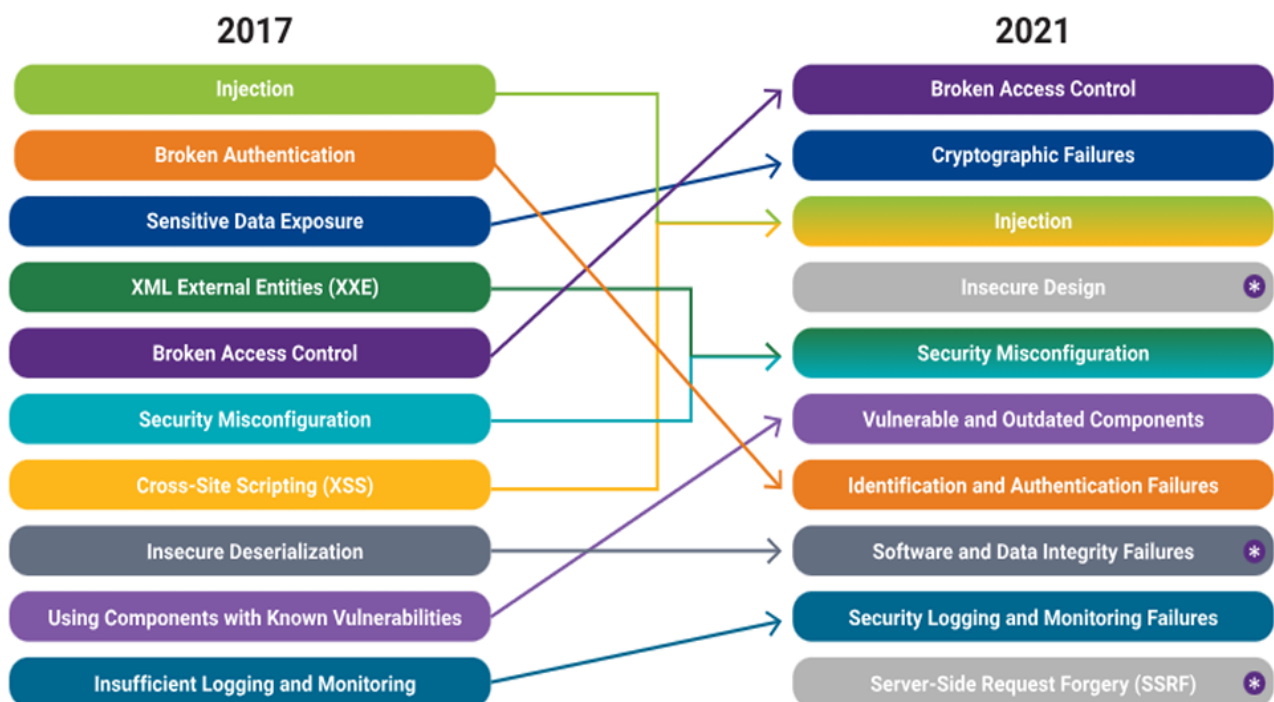


Fig1.1: OWASP Top 10 (2017 Vs 2021)

Reference Link: <https://www.synopsys.com/glossary/what-is-owasp-top-10.html>

SCOPE AND OBJECTIVES

The scope of this project involves a comprehensive evaluation of the digital infrastructure, web applications, and data protection mechanisms supporting the <https://www.jjmcoe.ac.in/> platform. The primary focus is on identifying vulnerabilities, weaknesses, and potential threats that could compromise the confidentiality, integrity, and availability of the website's resources. This security audit will encompass both technical and operational aspects, including an in-depth assessment of the web application's software, network architecture, user access controls, and adherence to relevant security standards and best practices. The assessment will be performed using a combination of manual techniques and automated scans with the latest, legitimate tools available in the cybersecurity domain.

The audit will also extend to evaluating user authentication mechanisms, encryption practices, and incident response procedures to ensure the robustness of the platform's security. Given that the audit will be conducted entirely online, the primary focus will be on the platform's online security, including the identification and mitigation of potential attack vectors.

The main objectives of the project are:

- To conduct a thorough assessment of the website's infrastructure to identify security vulnerabilities such as SQL injection, cross-site scripting (XSS), file upload vulnerabilities, weak password policies, and more.
- To provide actionable recommendations and best practices to remediate identified vulnerabilities and strengthen the overall security posture of the website.
- To enhance the website's resilience against cyber threats, safeguard user data, and build user trust by ensuring the platform operates securely and reliably.

By addressing these objectives, the project aims to improve the reputation and security of the <https://www.jjmcoe.ac.in/> website while ensuring a safe and secure experience for its users.

METHODOLOGY

The current security systems need to be tested for both substantive and compliance aspects. Compliance testing is done to assess whether controls are being applied according to the documentation offered by the client. It also checks if IT controls follow the compliance levels in accordance with management procedures and policies. In substantive testing, the adequacy of the controls is substantiated by whether they are able to protect the organization from cyber threats. These tests need an in-depth understanding of the different kinds of threats such as unauthorized access to assets including data, unusual interactions with the system, data corruption, inaccuracy in information, etc. Application controls are application-specific controls and have a high impact on individual transactions. These controls ensure and verify that all transactions are authorized, safe, and recorded. To proceed with this phase of the audit, there is a need for a deep understanding of the working of the system. For this analysis, a brief description of the application is required, along with details of transactions including volume, involved data, and flow. Most organizations either use local area networks for their operations. This leads to the risk of access by unauthorized users if not monitored and protected properly. The fundamental requirement of a network is to be accessible by only authorized users. Controls should be implemented to eliminate issues like data corruption, data loss, or interception while being transmitted.

IT Audit standards

The IT audit should comply with internationally accepted security standards. Some of these are

mentioned below:

- **ISO Compliance:** The ISO publishes a slew of guidelines that ensure reliability, quality, and safety. ISO 27001 is suitable for information security requirements.
- **PCI DSS Compliance:** These standards apply to any company that is involved with customer payments. This is necessary to ensure that all transactions are secure and protected.

PHASES OF SECURITY AUDIT

There are 4 significant phases in a security audit:

1. Planning phase

Preliminary information gathering and assessment

Planning is an integral part of any audit. In the beginning, planning is done to create a process flow based on an initial reconnaissance of the entire system. The plan is updated according to the test results of the initial assessment.

2. Audit scope and objective

From the above steps, the auditor gains relevant information and details to define the objective and scope of the audit in a clear and detailed format. The initial risk assessment forms an important part of the process and answers questions pertaining to three primary security goals, confidentiality, integrity, and reliability.

Risk assessment consists of ranking the potential threats from low to high, or other scientific or complex metrics. The ranking depends on the severity of the issue with respect to the extent of damage it can cause or the ease of exploitation. Vulnerabilities that are easy to exploit and those causing a high degree of damage must be ranked comparatively higher.

3. Evaluating collected evidence

Through rigorous testing and prodding of the security infrastructure, various types of evidence are gathered that must be interpreted to compile the results of the audit. There are various techniques to test a system and obtain results. Evidence can be majorly 3 types:

- ☐ Documentary evidence
- ☐ System analysis
- ☐ Observation of processes

4. Documenting audit results

Proper documentation of the results forms an integral part of security audit methodology. The final report should be in a very consumable format for stakeholders at all levels to understand and interpret. It must contain details such as the audit plan, audit scope, tests carried out, findings and detailed solutions, and next steps to remedy the security issues.

RECONNAISSANCE

Reconnaissance is the information-gathering stage of ethical hacking, where you collect data about the target system. This data can include anything from network infrastructure to employee contact details. The goal of reconnaissance is to identify as many potential attack vectors as possible.

How Reconnaissance Works

Reconnaissance generally follows seven steps:

1. Collect initial information
2. Determine the network range
3. Identify active machines
4. Find access points and open ports
5. Fingerprint the operating system
6. Discover services on ports
7. Map the network

TOOLS

Following are the tools which we used to perform reconnaissance on website “ktu.edu.in” -

1) DNSdumpster

DNSdumpster is an online passive scanning tool to obtain information about domains, block addresses, emails, and all kind of information DNS related. It is a tool to perform DNS reconnaissance on target networks. The results include a variety of information that are useful for users performing network reconnaissance. Some of the information return include

- Host subdomains
- Different dns informat (MX, A record)
- Geo information
- Email

It is an open source intelligence for the networks of your choice. With the help of this site or platform, the years can identify the attack surface or potentials. You can also analyze the security strategy related to the information of the network with the help of passive DNS reconnaissance, and eventually get rid of the threat to security. Since it's a web based service we only need to navigate to their url and query our target.

2) Pentest Tools (Online tool for website Vulnerability)

<https://pentest-tools.com/website-vulnerability-scanning/website-scanner>

Used this online tool for getting some preliminary information about the website as follows and here are the findings as follows.

IP Information

CONFIRMED

IP Address	Hostname	Location	Autonomous system (AS) Information	Organization (Name & Type)
148.135.138.72	www.jjmcoe.ac.in	Mumbai, Maharashtra, India	Amazon Inc (AS16509)	Amazon Inc (business)

DNS Records

CONFIRMED

Domain Queried	DNS Record Type	Description	Value
www.jjmcoe.ac.in	A	IPv4 address	148.135.138.72
www.jjmcoe.ac.in	NS	Name server	ns2.jjmcoe.ac.in
www.jjmcoe.ac.in	NS	Name server	ns1.jjmcoe.ac.in
www.jjmcoe.ac.in	MX	Mail server	5 alt2.aspmx.l.google.com
www.jjmcoe.ac.in	MX	Mail server	5 alt1.aspmx.l.google.com
www.jjmcoe.ac.in	MX	Mail server	1 aspmx.l.google.com
www.jjmcoe.ac.in	MX	Mail server	10 alt3.aspmx.l.google.com
www.jjmcoe.ac.in	MX	Mail server	10 alt4.aspmx.l.google.com
www.jjmcoe.ac.in	SOA	Start of Authority	ns2.jjmcoe.ac.in. email.jjmcoe.ac.in. 2024112844 10800 3600 1209600 10800
www.jjmcoe.ac.in	TXT	Text record	"google-site-verification=g8sAWbeiuWXqBIGLFIFECIctUJ3OvIWxrdYkhQWmiXE"
www.jjmcoe.ac.in	SPF	Sender Policy Framework	"v=spf1 include:_spf.google.com ~all"
www.jjmcoe.ac.in	CNAME	Canonical name	jjmcoe.ac.in

Risk Description















An initial step for an attacker aiming to learn about an organization involves conducting searches on its domain names to uncover DNS records associated with the organization. This strategy aims to amass comprehensive insights into the target domain, enabling the attacker to outline the organization's external digital landscape. This gathered intelligence may subsequently serve as a foundation for launching attacks, including those based on social engineering techniques. DNS records pointing to services or servers that are no longer in use can provide an attacker with an easy entry point into the network.

Recommendation

We recommend reviewing all DNS records associated with the domain and identifying and removing unused or obsolete records.

Server software and technologies

port 443/tcp

Software / Version	Category
 WordPress	CMS, Blogs
 MySQL	Databases
 PHP 8.2.26	Programming languages
 YouTube	Video players
 Nginx	Web servers, Reverse proxies
 Stimulus	JavaScript frameworks
 Site Kit 1.140.0	Analytics, WordPress plugins
 jQuery Migrate 3.4.1	JavaScript libraries
 jQuery	JavaScript libraries
 HSTS	Security
 Underscore.js 1.13.7	JavaScript libraries
 Clipboard.js	JavaScript libraries
 Priority Hints	Performance
 RSS	Miscellaneous

Risk Description

We noticed that server software and technology details are exposed, potentially aiding attackers in tailoring specific exploits against identified systems and versions. The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation

We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

❖ NMAP

Nmap is short for Network Mapper. It is an open-source Linux command-line tool that is used to scan IP addresses and ports in a network and to detect installed applications. Nmap allows network admins to find which devices are running on their network, discover open ports and services, and detect vulnerabilities.

Why use Nmap?

There are a number of reasons why security pros prefer Nmap over other scanning tools.

- First, Nmap helps you to quickly map out a network without sophisticated commands or configurations. It also supports simple commands (for example, to check if a host is up) and complex scripting through the Nmap scripting engine.
- Ability to quickly recognize all the devices including servers, routers, switches, mobile devices, etc on single or multiple networks.
- Helps identify services running on a system including web servers, DNS servers, and other common applications. Nmap can also detect application versions with reasonable accuracy to help detect existing vulnerabilities.
- Nmap can find information about the operating system running on devices. It can provide detailed information like OS versions, making it easier to plan additional approaches during penetration testing.

1. Service Version Detection

Command:

nmap -sV www.jjmcoe.ac.in

```

(root@kali)~[/home/kali]
# nmap -sV www.jjmcoe.ac.in
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-02 06:15 EST
Nmap scan report for www.jjmcoe.ac.in (148.135.138.72)
Host is up (0.034s latency).
rDNS record for 148.135.138.72: vps.happy-visitors.com
Not shown: 989 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD
22/tcp    open  ssh          OpenSSH 8.0 (protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       (unknown banner: none)
80/tcp    open  http         nginx
106/tcp   open  tcpwrapped
110/tcp   open  pop3         Dovecot pop3d
143/tcp   open  imap         Dovecot imapd
443/tcp   open  ssl/http     nginx
993/tcp   open  ssl/imap     Dovecot imapd
995/tcp   open  ssl/pop3     Dovecot pop3d
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.94SVN%I=7%D=1/2%Time=67767571%P=x86_64-pc-linux-gnu%r(DN
SF:SVersionBindReqTCP,3F,"0=\0\x06\x85\0\0\x01\0\x01\0\x01\0\0\x07version
SF:\x04bind\0\0\x10\0\03\0c\0\x10\0\03\0\0\0\0\05\x04none\0c\0\x0
SF:c\0\02\0\03\0\0\0\0\02\0c\0c");
Service Info: Host: vps.happy-visitors.com

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.29 seconds

```

Port	State	Service	Version	Remarks
21/tcp	Open	FTP	ProFTPD	May allow unauthorized access or brute force attacks, Ensure FTP is secured.
22/tcp	Open	SSH	OpenSSH 8.0 (Protocol 2.0)	SSH version is outdated. Ensure strong authentication mechanisms are in place.
25/tcp	Open	SMTP	Postfix smtpd	Potential target for email spamming or spoofing.
53/tcp	Open	DNS	Unknown banner	DNS service running. Ensure no sensitive zone transfer is allowed.
80/tcp	Open	HTTP	nginx	Ensure all HTTP services are redirected to HTTPS for secure communication.
110/tcp	Open	POP3	Dovecot imapd	Ensure proper encryption for IMAP to prevent data leaks.
443/tcp	Open	HTTPS	nginx	Ensure SSL/TLS is properly configured to avoid vulnerabilities.
993/tcp	Open	Secure IMAP	Dovecot imapd	Confirm proper implementation of

				secure email protocols.
995/tcp	Open	Secure POP3	Dovecot pop3d	Secure service, but confirm proper configuration.
8443/tcp	Open	HTTPS-alt	sw-cp-server	Confirm service usage and ensure proper security measures.

Potential Risks Identified

Potential Risks Identified

1. **FTP (Port 21):** May allow attackers to brute force credentials or exploit unencrypted file transfers.
2. **Outdated SSH (Port 22):** Running OpenSSH 8.0, which might be vulnerable to known exploits.
3. **HTTP (Port 80):** Traffic is unencrypted, exposing data to interception. HTTP-to-HTTPS redirection is recommended.
4. **Email Services (Ports 25, 110, 143):** Improperly secured email protocols could lead to email spoofing or credential theft.
5. **DNS Service (Port 53):** May allow unauthorized zone transfers or information leakage.
6. **SSL/TLS (Ports 443, 993, 995, 8443):** Ensure no deprecated protocols (e.g., TLS 1.0/1.1) are in use and SSL certificates are valid.

Recommendations

- Restrict or secure FTP access using SFTP or disabling FTP altogether if not required.
- Update OpenSSH to the latest secure version and implement key-based authentication.
- Enforce HTTPS for all web traffic with a valid SSL certificate.
- Disable plaintext email services (POP3/IMAP) and enforce encryption.
- Validate and secure DNS configurations to prevent unauthorized zone transfers.
- Conduct further vulnerability assessments for each open port to identify specific risks.

❖ NIKTO

Nikto is an open source web server and web application scanner. Nikto can perform comprehensive tests against web servers for multiple security threats, including over potentially dangerous files/programs. Nikto can also perform checks for outdated web servers software, and version-specific

Here are some of the things that Nikto can do:

- Find SQL injection, XSS, and other common vulnerabilities
- Identify installed software (via headers, favicons, and files)
- Guess subdomains
- Includes support for SSL (HTTPS) websites
- Saves reports in plain text, XML, HTML or CSV
- Guess credentials for authorization (including many default username/password combinations)problems.

The `nikto` tool has been used to perform a security scan on the domain “<https://www.jjmcoe.ac.in/>”. `Nikto` is a web server scanner that identifies potential vulnerabilities and security issues on web servers by sending a series of requests and analyzing the responses.

The scan results provide insights into the security posture of the web server associated with the provided domain name. `Nikto` conducts a comprehensive scan that includes checks for known vulnerabilities, outdated software, misconfigurations, and potential security risks.

The output of the scan may include information about open ports, discovered directories, files, and server-specific issues. It can also highlight potential security vulnerabilities such as outdated software versions, insecure configurations, or exposed sensitive information.

It's important to note that `Nikto` is a tool that helps identify potential security issues; it doesn't guarantee the presence of vulnerabilities. The results should be carefully reviewed and verified, and any identified issues should be further investigated and addressed.

In summary, the `nikto` scan on the domain <https://www.jjmcoe.ac.in/> aims to uncover potential security vulnerabilities and misconfigurations on the web server. The results will aid administrators and security professionals in understanding the current security state of the web server and taking appropriate measures to mitigate any identified risks.

Here are the results of the Nikto scan and our findings based on the result.

```
(root@kali)-[/home/kali]
# nikto -h https://www.jjmcoe.ac.in
- Nikto v2.5.0

+ Target IP: 148.135.138.72
+ Target Hostname: www.jjmcoe.ac.in
+ Target Port: 443

+ SSL Info: Subject: /CN=jjmcoe.ac.in
            Ciphers: TLS_AES_256_GCM_SHA384
            Issuer: /C=US/O=Let's Encrypt/CN=R11
+ Start Time: 2025-01-02 04:44:20 (GMT-5)

+ Server: nginx
+ /: Retrieved x-powered-by header: PHP/8.2.26, PleskLin.
+ /: Drupal Link header found with value: <https://www.jjmcoe.ac.in/wp-json/; rel="https://api.w.org/", <https://www.jjmcoe.ac.in/wp-json/wp/v2/pages/634>; rel="alternate"; title="JSON"; type="application/json", <https://www.jjmcoe.ac.in/; rel=shortlink. See: https://www.drupal.org/
+ /930ap28E.LOG: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /930ap28E.: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
+ Hostname 'www.jjmcoe.ac.in' does not match certificate's names: jjmcoe.ac.in. See: https://cwe.mitre.org/data/definitions/297.html
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /download/: This might be interesting.
+ /error_log: This might be interesting.
+ /img/: This might be interesting.
+ /staff/: This might be interesting.
```

```
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /bs/: This might be interesting: potential country code (Bahamas).
+ /tm/: This might be interesting: potential country code (Turkmenistan).
+ /: A Wordpress installation was found.
+ /wordpress/: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Wordpress login found.
+ /.well-known/carddav: CardDAV file may contain server info. See: RFC-5785 http://tools.ietf.org/html/rfc6764
+ 8973 requests: 0 error(s) and 20 item(s) reported on remote host
+ End Time: 2025-01-02 07:40:30 (GMT-5) (7331 seconds)
```

```
+ 1 host(s) tested
```

```
(root@kali)-[/home/kali]
#
```

Based on the provided information, We will break down each identified vulnerability, including its impact, CVSS score (estimated), CWE ID, and OWASP category:

1. X-Powered-By Header Exposure

Impact:

Exposing the X-Powered-By header (PHP/8.2.26, PleskLin) discloses the backend technology and its version. This could allow attackers to:

Identify the server software and target known vulnerabilities in PHP 8.2.26 or Plesk.

Exploit misconfigurations or unpatched vulnerabilities.

CVSS Score: 4.0 (Medium)

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

CWE ID:CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

OWASP:A01:2021 - Broken Access Control (Information Disclosure)

2. Missing X-Content-Type-Options Header

Impact:

Without the X-Content-Type-Options header, browsers may perform MIME sniffing, potentially rendering content in an unsafe manner. This can lead to:

Cross-Site Scripting (XSS): Injected scripts could be executed if the browser misinterprets content.

MIME-based attacks: An attacker could exploit file uploads or misconfigured resources.

CVSS Score:6.1 (Medium)

Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N

CWE ID:CWE-116: Improper Encoding or Escaping of Output

OWASP:A03:2021 - Injection (XSS risk due to MIME sniffing)

3. Uncommon Header (x-redirect-by: WordPress)

Impact:

Revealing the header x-redirect-by indicates the site is running WordPress. This information could help attackers:

Target known vulnerabilities in the WordPress core or plugins.

Tailor attacks based on WordPress-specific exploits.

CVSS Score:4.0 (Medium)

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

CWE ID:CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

OWASP:A01:2021 - Broken Access Control (Information Disclosure)

4. Mismatched Hostname in SSL Certificate

Impact:

The hostname www.jjmcoe.ac.in does not match the SSL certificate's Common Name (jjmcoe.ac.in). This could:

Trigger warnings in browsers, leading to a loss of trust from users.

Allow attackers to perform Man-in-the-Middle (MITM) attacks by exploiting the mismatch.

CVSS Score: 6.4 (Medium)

Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:L

CWE ID:CWE-297: Improper Validation of Certificate with Host Mismatch

OWASP:A02:2021 - Cryptographic Failures

5. Robots.txt Contains Sensitive Entries

Impact:

The robots.txt file contains entries that may expose sensitive paths or files to attackers. While not directly exploitable, it could:

Allow attackers to identify restricted or administrative directories.

Lead to further enumeration and potential exploitation.

CVSS Score: 3.7 (Low)

Vector: AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

CWE ID:CWE-538: File and Directory Information Exposure

OWASP:A06:2021 - Security Misconfiguration

6. Junk HTTP Methods Return Valid Response

Impact:

When the server responds with valid responses to junk HTTP methods, it may:

Indicate misconfigured HTTP method handling.

Allow attackers to attempt unconventional HTTP requests to bypass security measures or exploit undocumented behavior.

CVSS Score: 5.3 (Medium)

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

CWE ID: **CWE-670:** Always-Incorrect Control Flow Implementation

OWASP: **A06:2021** - Security Misconfiguration.

1) Exposure of Faculty contact numbers and Emails

IMPACT	HIGH(indirect) Attackers could use this information for phishing, social engineering, or spam campaigns, potentially leading to data breaches, unauthorized access, or reputational damage for faculty and the institution.
CWE	CWE-200: Exposure of Sensitive Information to an unauthorised Actor. CWE-523: Unprotected Transport of Credentials (if contact details are used for further exploitation).
OWASP	A06:2021 – Vulnerable and Outdated Components(if publicly accessible directories are a result of misconfigured or outdated CMS) A01:2021 – Broken Access Control(If this data should have been restricted to authorised users only)
CVSS SCORE:	7.5

Vulnerability Description

During the security audit of <https://engineering.jjm.ac.in>, it was found that faculty email addresses and phone numbers are publicly accessible on the website. This exposure poses risks such as phishing, social engineering attacks, spam campaigns, and unauthorized profiling of faculty members for targeted attacks.

Limiting public access to this information is crucial to prevent potential misuse and safeguard the organization's reputation and security.

How it works

In About Section > Committee -list , committee list from 2022 to 2025 are mentioned with detailed information about all the members with the phone numbers and their emails.

The availability of this information on a public-facing platform makes it vulnerable to harvesting by attackers, who could use it for phishing campaigns, social engineering attacks, or spam distribution. These attacks can lead to credential theft, unauthorized access to sensitive information, or even reputational damage to both the institution and its faculty members.

Additionally, publicly accessible email addresses and phone numbers can serve as valuable data points for attackers performing reconnaissance, enabling them to map the organization's hierarchy and identify key personnel for targeted attacks, such as spear-phishing or Business Email Compromise (BEC).

This finding underscores the importance of limiting the exposure of sensitive or personally identifiable information (PII) on public websites to safeguard against indirect attack vectors that could compromise the confidentiality, integrity, and trustworthiness of the organization's digital ecosystem.

Proof of Concept

After visiting one of the committee list , some of the proofs are attaching here.

ijmcoe.ac.in/wp-content/uploads/2024/11/ALL-COMM-FINAL-2024-25-WEBSITE.pdf

6 / 32 | 100% + | [] []

ALL-COMM-FINAL-2024-25-WEBSITE.pdf

Dr.J.J.MagdumTrust's
Dr. J. J. Magdum College of Engineering, Jaysingpur.
(An Autonomous Institute)

Equipment / Purchase Committee (2024-25)

Sr.No.	Name	Dept.	Type of Nomination	Mob.No.	E-Mail
1.	Dr. D. B. Unde	Gen Engg.	Head	9764440009	declip.unde@ijmcoe.ac.in
2.	Dr. J. S. Lambe	CivilEngg.	Member	9421204424	hodcivil@ijmcoe.ac.in
3.	Prof. M. M. Kolap	ETC Engg.	Member	9273961061	hodetc@ijmcoe.ac.in
4.	Prof. Mrs. M. U. Phutane	ETC	Member	7709904600	manisha.phutane@ijmcoe.ac.in
5.	Prof. R.A.Bharatiya	IT	Member	9860650444	hodit@ijmcoe.ac.in
6.	Prof. M. B. Bhilavade	Gen Engg..	Member	9420675861	hodfe@ijmcoe.ac.in
7.	Mr. D. R. Mane	Librarian	Member	8637795321	librarian@ijmcoe.ac.in
8.	Mr. J. B. Patil	Accountant	Member Secretary	9881345366	accounts@ijmcoe.ac.in
9.	Mr. S. S. Supanekar	Storekeeper	Member	9765346469	stores@ijmcoe.ac.in

ijmcoe.ac.in/wp-content/uploads/2024/11/ALL-COMM-FINAL-2024-25-WEBSITE.pdf

7 / 32 | 100% + | [] []

ALL-COMM-FINAL-2024-25-WEBSITE.pdf

Dr.J.J.MagdumTrust's
Dr. J. J. Magdum College of Engineering, Jaysingpur.
(An Autonomous Institute)

Finance Committee (2024-25)

Sr.No.	Name	Dept.	Type of Nomination	Mob.No.	E-Mail
1.	Dr.Sunil Admuthe	Campus Director	Chairman	9422411441	campusdirector@ijmcoe.ac.in
2.	Dr. G. V. Mulgund	Principal	Member	7588839791	principal@ijmcoe.ac.in
3.	Dr. D. B. Unde	Gen Engg.	Head	9764440009	declip.unde@ijmcoe.ac.in
4.	Prof.S.T. Jadhav	Registrar	Member	9823204850	registrar@ijmcoe.ac.in
5.	Mr. J.B.Patil	Finance officer	Member	9881346653	accounts@ijmcoe.ac.in
6.	Prof. B.N.Shinde	F.Y.B.Tech.	Member Secretary	9421175569	Balaram.shinde@ijmcoe.ac.in

Recommendation

1. **Restrict Public Access:** Remove email addresses and phone numbers from publicly accessible sections of the website. Instead, implement a secure contact form for communication.
2. **Implement Email Obfuscation:** If contact details must be displayed, use obfuscation techniques (e.g., "name [at] domain [dot] com") or JavaScript-based methods to prevent automated harvesting.
3. **Enable CAPTCHA on Forms:** Add CAPTCHA to contact forms to mitigate automated bot abuse and spam submissions.
4. **Raise Awareness:** Train faculty members to recognize phishing and social engineering attempts. Provide guidelines on responding to suspicious emails or calls.
5. **Regular Monitoring:** Continuously monitor for unauthorized access or abnormal activities related to faculty accounts. Deploy security alerts for any suspicious behavior.

2) Accessible Error Logs

IMPACT	HIGH (Expose to sensitive information)
CWE	CWE-532: Information Exposure Through log Files
OWASP	A06:2021 – Security Misconfiguration
CVSS SCORE:	7.5

Vulnerability Description

The website **/error-logs** endpoint reveals detailed error logs, which may disclose sensitive information such as server configurations, file paths, software versions, database queries, or stack traces. This exposure can aid attackers in understanding the application's internal structure, facilitating exploits like SQL injection, remote code execution, or authentication bypass. Additionally, it increases the risk of targeted attacks and diminishes the organization's security posture.

How it Works

In General:

Error logs capture detailed information about system or application errors that occur during the operation of a website or application. These logs often include:

- **Stack traces** (detailing error locations in the code).
- **Database queries** (which can reveal sensitive database structures or data).
- **Server configurations** (such as paths or environment settings).
- **Software versions** (indicating which frameworks or libraries are being used).

When these logs are not properly secured, they may become accessible to anyone who visits the site or scans for them. This gives attackers valuable insight into the internal workings of the system and makes it easier for them to craft targeted attacks.

For Our Case (on <https://engineering.jjm.ac.in>):

In the case of <https://engineering.jjm.ac.in>, the **/error-logs** path exposes error logs that may contain:

1. **Duplicate Constant Definition:**
The warning about WP_CACHE being redefined suggests a misconfiguration in the wp-config.php file, where the constant is defined multiple times, leading to redundancy and potential conflicts.
2. **Improper Method Visibility:**
Warnings about the magic method __wakeup() in the Visual_Form_Builder plugin lacking proper visibility indicate coding standards issues, as certain methods require specific visibility

Proof of Concept

After going to the path /error-logs, it is showing the direct logs as follows:

```
← → ↺ 📄 https://www.jjmcoac.in/error_log ⓘ ☆
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

26-Aug-2022 17:53:49 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:53:50 UTC] PHP Warning: The magic method Visual Form Builder:: wakeup() must have public visibility in /home2/jjmcoac/public_html/wp-content/plugins/visual-form-builder/visual-form-builder.php on line 68
26-Aug-2022 17:53:50 UTC] PHP Warning: The magic method Visual Form Builder Form Display:: wakeup() must have public visibility in /home2/jjmcoac/public_html/wp-content/plugins/visual-form-builder/public/class-form-display.php on line 35
26-Aug-2022 17:53:52 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:53:53 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:54:55 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:54:55 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:54:57 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:56:54 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:56:56 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:57:29 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:57:30 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:58:41 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:58:41 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:58:42 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:59:05 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 17:59:06 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:05 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:06 UTC] PHP Warning: The magic method Visual Form Builder:: wakeup() must have public visibility in /home2/jjmcoac/public_html/wp-content/plugins/visual-form-builder/visual-form-builder.php on line 68
26-Aug-2022 18:02:06 UTC] PHP Warning: The magic method Visual Form Builder Form Display:: wakeup() must have public visibility in /home2/jjmcoac/public_html/wp-content/plugins/visual-form-builder/public/class-form-display.php on line 35
26-Aug-2022 18:02:06 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:07 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:08 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:09 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:54 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:02:56 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:03:22 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:03:23 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:03:36 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:03:37 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:06:47 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:06:48 UTC] PHP Warning: The magic method Visual Form Builder:: wakeup() must have public visibility in /home2/jjmcoac/public_html/wp-content/plugins/visual-form-builder/visual-form-builder.php on line 68
26-Aug-2022 18:06:48 UTC] PHP Warning: The magic method Visual Form Builder Form Display:: wakeup() must have public visibility in /home2/jjmcoac/public_html/wp-content/plugins/visual-form-builder/public/class-form-display.php on line 35
26-Aug-2022 18:06:50 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:07:17 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:07:18 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:07:35 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:07:36 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:07:37 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:07:45 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:08:31 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:08:33 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:10:02 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:11:23 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:11:24 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:11:53 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:11:55 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:12:16 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:12:21 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:12:23 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:12:25 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:12:27 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:13:19 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
26-Aug-2022 18:13:30 UTC] PHP Warning: Constant WP_CACHE already defined in /home2/jjmcoac/public_html/wp-config.php on line 51
```

Recommendation

1. Fix Redundant Definitions:

Check the wp-config.php file and remove any duplicate or unnecessary definitions of WP_CACHE.

2. Update Plugins:

Ensure all plugins, including Visual_Form_Builder, are updated to their latest versions.

3. Code Review:

Examine plugin/theme files for coding standard adherence, especially for PHP magic methods.

4. Testing After Changes:

Always test changes on a staging environment before applying them to the live system.

3)Weak Encoding of Credentials (Base64 Encoding)

IMPACT

HIGH (Expose to sensitive information)

CWE

CWE-532: Missing Encryption of Sensitive Data.

OWASP

A02:2021 – Cryptographic Failures

CVSS SCORE:

9.1

Vulnerability Description

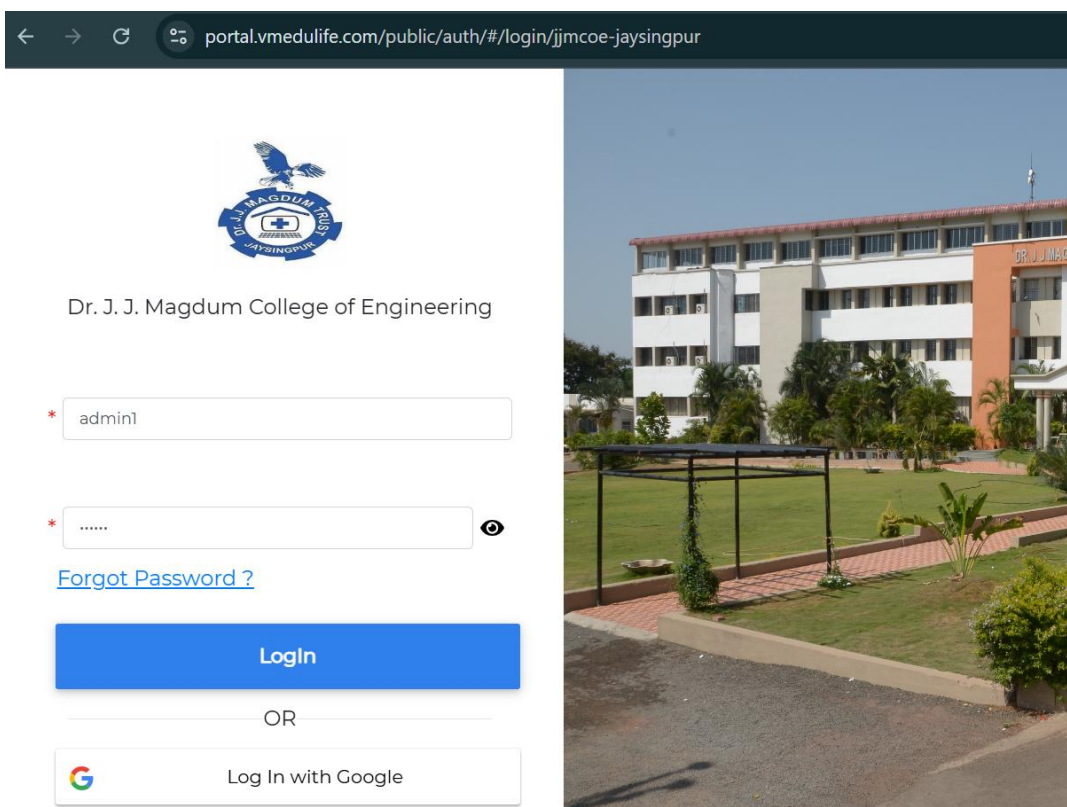
The login request was intercepted, revealing that the credentials were transmitted in **Base64 encoding**. Base64 is not an encryption method but merely an encoding scheme that can be easily decoded. As a result, an attacker can decode the encoded credentials using readily available tools, exposing sensitive login information in plaintext.

How it Works?

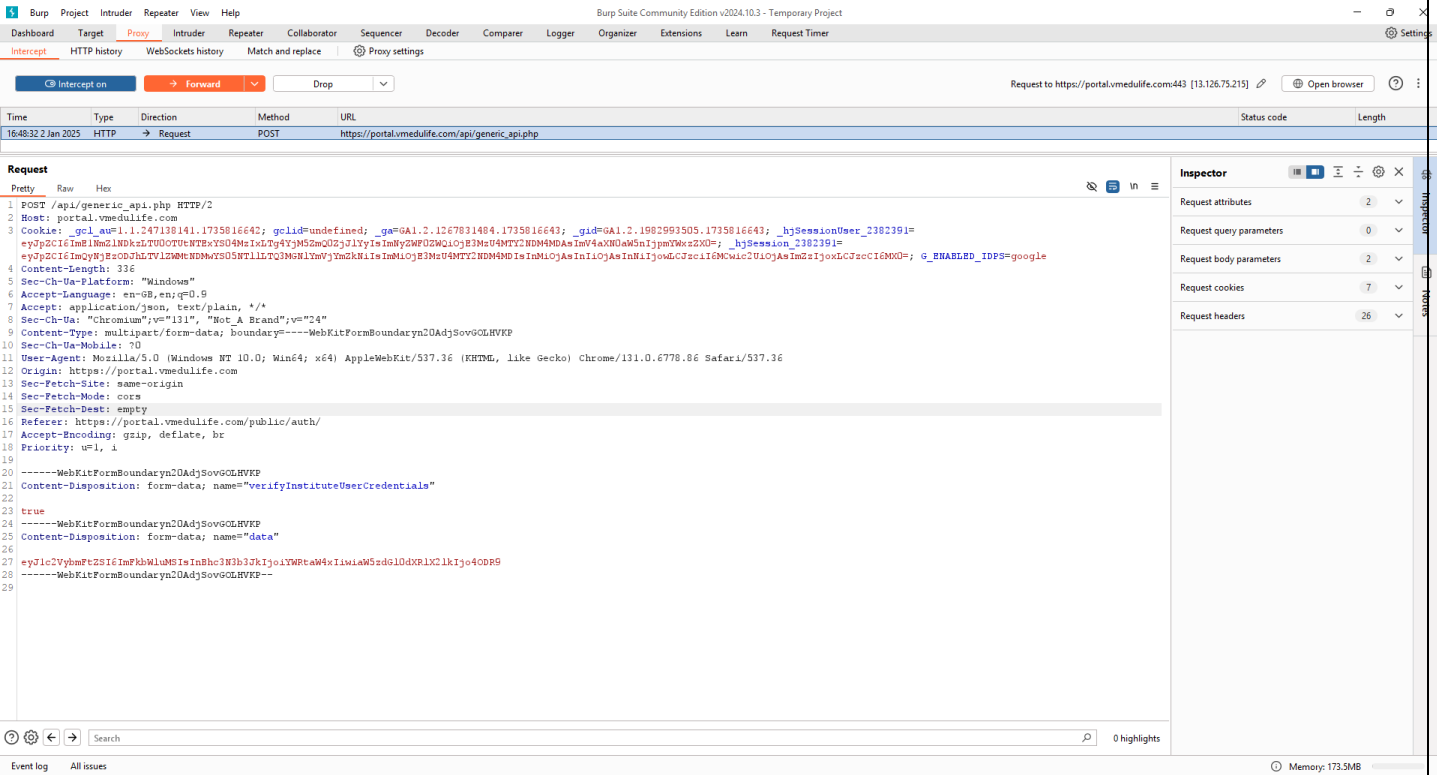
- 1. Credentials Exposure:** Attackers can intercept and decode Base64-encoded credentials, gaining unauthorized access.
- 2. Man-in-the-Middle (MITM) Attacks:** If the transmission is not encrypted (e.g., no HTTPS), credentials can be intercepted over the network.
- 3. Privilege Escalation:** If an attacker gains access to an administrative account, they can compromise the entire application.

Proof of Concept

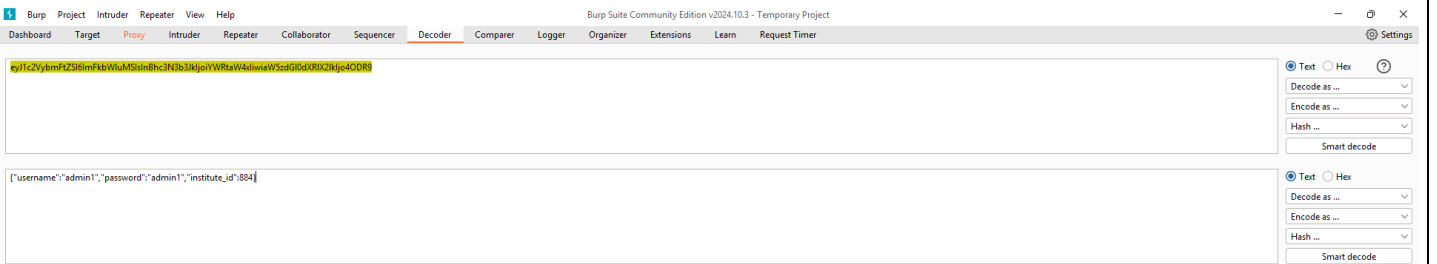
First we go to the login page and type any username and password



Then started the proxy on the browser to intercept the request in Burp suite and click on login button



Using Decoder in burp suite decoded the last line of this intercepted request and we get the same details that we tried to login



Recommendation

- 1. Use Strong Encryption:** Transmit credentials over HTTPS to ensure data is encrypted in transit.
- 2. Avoid Base64 Encoding for Sensitive Data:** Use secure hashing (e.g., bcrypt, Argon2) for storing passwords and strong encryption (e.g., AES) for transmitting sensitive data.
- 3. Enable Secure Communication:** Ensure SSL/TLS is properly configured and enforced across the application.
- 4. Implement Token-Based Authentication:** Use secure authentication tokens instead of transmitting raw credentials for every request.
- 5. Monitor and Audit Logs:** Regularly review access logs for any unauthorized attempts or anomalies.

4)Lack of Input Validation in Contact Us Form

IMPACT**HIGH****CWE**

CWE-79: Improper Neutralization of input During Web Page Generation (XSS)

CWE-20: Improper Input Validation

OWASP

A01-2021: Broken Access Control (Indirectly linked if malicious input bypass backend protections).

A03-2021: Injection (Lack of sanitization)

A05-2021: Security misconfiguration(Improper Handling of input Validation rules)

CVSS SCORE:

7.5-8.0 (Due to possibility of XSS)

Vulnerability Description

The Contact Us form does not enforce proper input validation, leading to several vulnerabilities:

1. **Name Field:** Accepts numbers instead of only alphabets or valid name formats.
2. **Mobile Number Field:** Accepts invalid numbers, such as 4-digit values or incorrectly formatted phone numbers.
3. **Email Field:** Does not validate email addresses using a proper regex pattern, allowing invalid or fake email addresses to pass through.
4. **Subject and Message Fields:** Accepting malicious payloads, such as `<script>alert(1)</script>`, indicates the potential for **Cross-Site Scripting (XSS)** attacks.

Impact:-**1. Cross-Site Scripting (XSS):**

- Attackers can inject malicious scripts, potentially compromising user data, session cookies, or executing unauthorized actions on the site.

2. Spam and Invalid Data:

- Lack of validation can lead to spam entries, overwhelming the backend or database with irrelevant or harmful data.

3. User Experience Degradation:

- Invalid data might create issues in downstream processes (e.g., email notifications or database storage).

4. Security Risks:

- Incomplete or malicious data could exploit other vulnerabilities in the application.

Proof of Concept

As per mentioned in the description, we filled the contact form in the following image.

Contact Form



Name *

1254784

Mobile No *

9999

Email Id *

12545@gmail.com

Subject

<script>alert(1)</script>

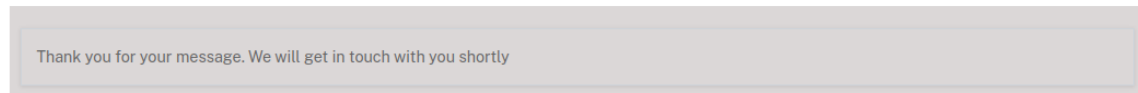
Your Message

<script>alert(1)</script>|

SUBMIT FORM

Accepting the form with the details we have mentioned.

Contact Form



Thank you for your message. We will get in touch with you shortly

Recommendation

1. Server-Side Input Validation

- Always validate user inputs on the server side, checking for data type, length, and format.
- Reject unexpected characters or inputs that don't match the expected format.

2. Client-Side Validation

- Use HTML5 attributes (required, pattern, maxlength) and JavaScript validation to provide quick feedback before submission.
- However, do not rely solely on client-side validation, as attackers can bypass it.

3. Sanitize and Escape Input Data

- Sanitize user inputs by removing unwanted characters.
- Escape special characters before displaying user input to prevent XSS attacks.

4. Use CAPTCHA or reCAPTCHA

- Implement CAPTCHA or reCAPTCHA to prevent automated bot submissions.

5. Restrict Special Characters and Apply Regex

- Limit special characters in fields like name and message.
- Use regex to validate input formats for email, phone numbers, and other fields.

5)Improper Access Control:- Verifying any emails in Grievance Redressal Portal

IMPACT	Medium (Allow only unauthorized grievance submission)
CWE	CWE-285: Improper Authorization → If grievance submission is meant only for students but allows anyone.
OWASP	A01:2021 - Broken Access Control → If non-students can verify their email and misuse the system.
CVSS SCORE:	6.5

Vulnerability Description

The grievance form of JJ College of Engineering allows **email verification for non-students**, bypassing intended access restrictions. This occurs due to improper authorization checks, allowing external users to verify their emails and potentially submit grievances.

How It Works:

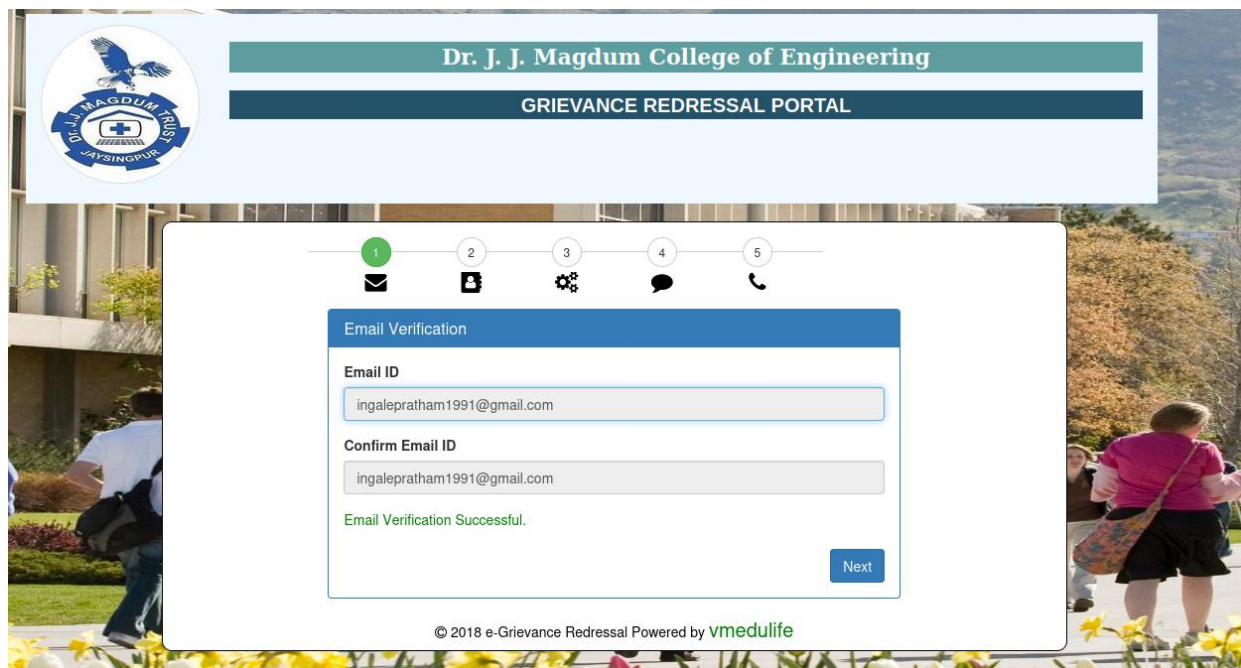
1. A non-student enters an email in the grievance form.
2. The system verifies the email without enforcing domain restrictions (e.g., @college.edu).
3. The non-student may gain access to grievance submission, impersonating students or misusing the system.

Impact:

- **Unauthorized grievance submissions**, leading to spam or system misuse.
- **Identity spoofing**, allowing impersonation of students.
- **Potential data leakage**, if verification responses expose sensitive information.

Proof of Concept

In the Grievance Redressal Portal section , if we put any email (Non-student) email , it is verifying this email successfully, we can see that in the following image.



Dr. J. J. Magdum College of Engineering

GRIEVANCE REDRESSAL PORTAL

1 2 3 4 5

Email Verification

Email ID

ingalepratham1991@gmail.com

Confirm Email ID

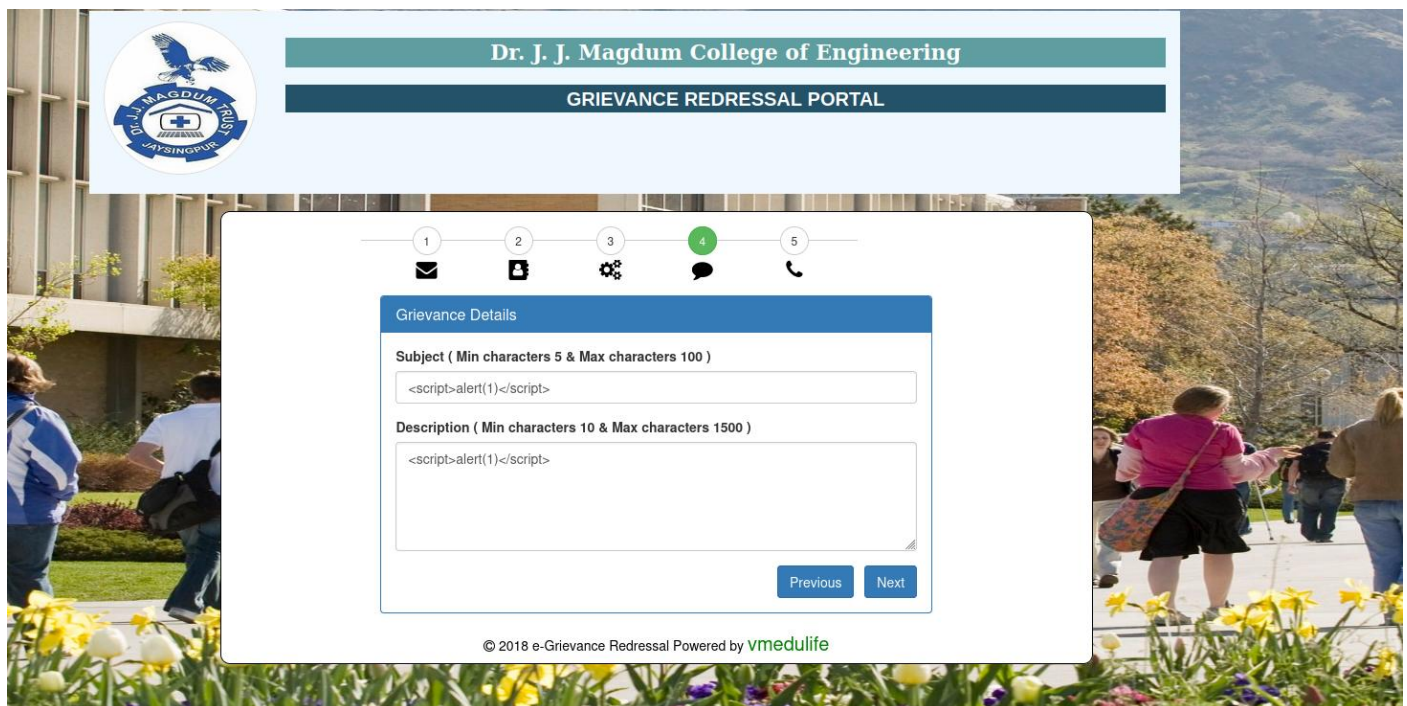
ingalepratham1991@gmail.com

Email Verification Successful.

Next

© 2018 e-Grievance Redressal Powered by vmedulife

Also in the Grievance details section, it is accepting the characters like <,/, (leading it to the cross site scripting kind of attacks.



Dr. J. J. Magdum College of Engineering

GRIEVANCE REDRESSAL PORTAL

1 2 3 4 5

Grievance Details

Subject (Min characters 5 & Max characters 100)

<script>alert(1)</script>

Description (Min characters 10 & Max characters 1500)

<script>alert(1)</script>

Previous Next

© 2018 e-Grievance Redressal Powered by vmedulife

Recommendation

- 1. Enforce Email Domain Restrictions** – Allow only official student emails (e.g., @college.edu) for verification.
- 2. Require Authentication** – Implement student login before accessing the grievance form.
- 3. Rate Limiting & CAPTCHA** – Prevent automated spam and abuse.
- 4. Identity Verification** – Cross-check emails with the student database before submission.
- 5. Improve Access Control** – Implement server-side validation and role-based access control (RBAC).
- 6. Monitor & Log Activity** – Track unusual grievance submissions and enable alerts for suspicious actions.

6) Brute Force Lockout mechanism not applied

IMPACT**High****CWE**

CWE-307: Improper Restriction of Excessive Authentication Attempts

OWASP

A07:2021 - Broken Access Control → Identifications and Authentication Failures

CVSS SCORE:

7.5

Vulnerability Description

Brute force attacks involve repeatedly attempting different username-password combinations to gain unauthorized access. If a system does not have a lockout mechanism, attackers can continuously try passwords without restriction, increasing the likelihood of a successful attack.

How It Works:

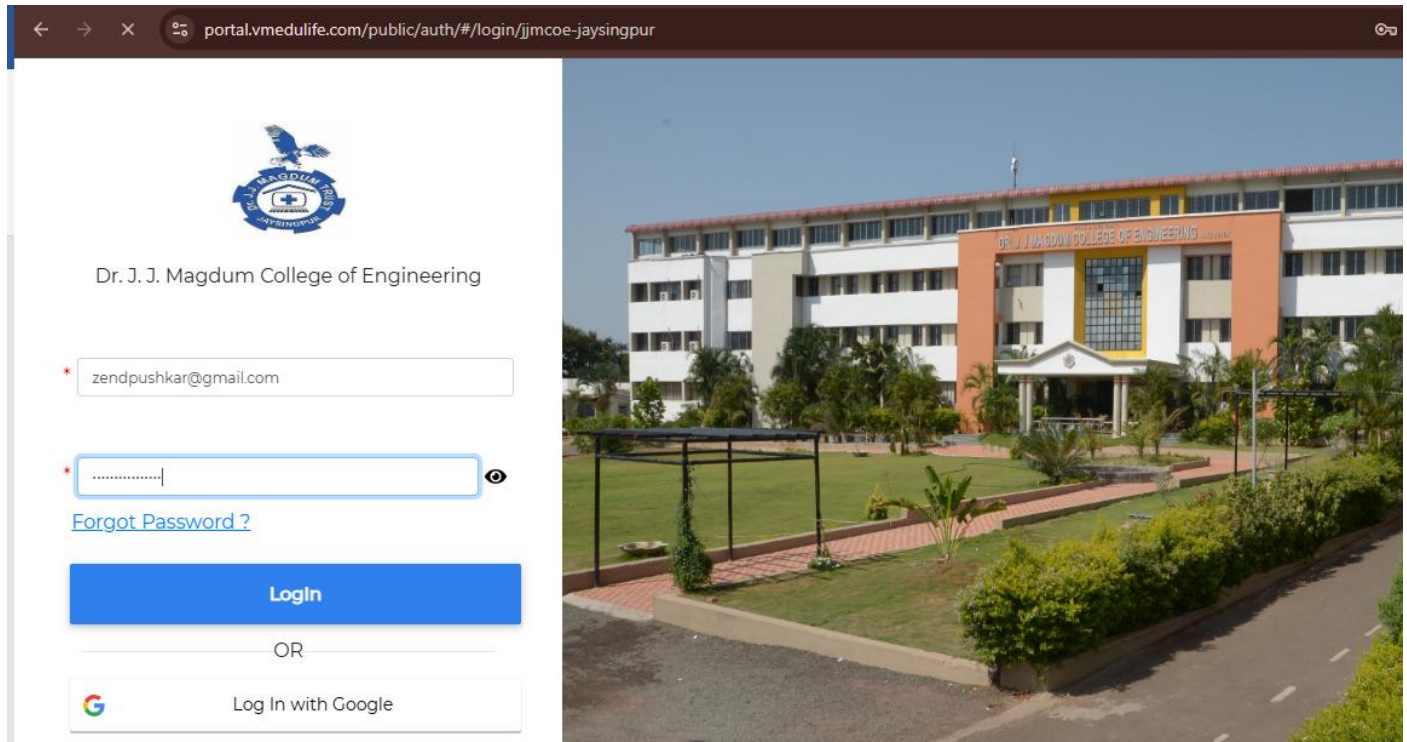
1. An attacker targets the login page and starts guessing credentials using automated tools like Hydra, Burp Suite Intruder, or Medusa.
2. Without a lockout mechanism, the attacker can make unlimited login attempts.
3. Using wordlists or credential stuffing techniques, they systematically try different combinations until a valid credential pair is found.
4. Once successful, the attacker gains unauthorized access to the account or system.

Impact:

- **Unauthorized account access** leading to data breaches.
- **Account takeover** if valid credentials are compromised.
- **Increased server load**, potentially leading to denial-of-service (DoS).

Proof of Concept

We tried to login with the same username multiple times with different passwords and it is not blocking that particular email id or username after more than 3 attempts.



Recommendation

- **Implement Account Lockout** – Temporarily lock accounts after multiple failed login attempts (e.g., 5-10 tries).
- **Introduce Progressive Delays** – Increase the delay between login attempts (e.g., 5s, 10s, etc.) to slow down attacks.
- **Enable CAPTCHA/ReCAPTCHA** – Require CAPTCHA after multiple failed attempts to block bots.
- **Block Suspicious IPs** – Use security tools like fail2ban or WAF to detect and block repeated login failures from the same IP.
- **Use Multi-Factor Authentication (MFA)** – Require an additional factor (OTP, email, app verification) after repeated failures.
- **Monitor and Alert** – Log failed attempts and alert administrators when unusual login activity is detected.

CONCLUSION

We conclude that the security audit of the website <https://www.jjmcoe.ac.in/> using the **OWASP Methodology** has been conducted through a systematic and thorough process to identify vulnerabilities and potential security threats.

The **OWASP Top 10** serves as a prioritized framework for assessing security risks, providing a baseline for website vulnerability testing. This approach enhances application security and reduces risks by implementing necessary mitigations against potential cyber threats and data breaches.

It is important to recognize that website security testing is an ongoing process. Since cybercriminals continuously develop sophisticated attack methods, achieving a fully secure website is nearly impossible. Therefore, **regular security audits—conducted annually or semi-annually—are essential** to protect the website from emerging cyber threats and ensure its continued resilience against evolving security risks.

REFERENCES

1. <https://owasp.org/www-project-top-ten/>
2. <https://cheatsheetseries.owasp.org/IndexTopTen.html>
3. <https://portswigger.net/web-security/cross-site-scripting>
4. <https://github.com/payloadbox/sql-injection-payload-list>
5. <https://www.kali.org/tools/nmap/>
6. <https://www.kali.org/tools/nikto/>
7. <https://www.kali.org/tools/sqlmap/>