



PROJECT REPORT

CYBERSECURITY  
  
WEB SERVICES OPEN TO MALICIOUS ATTTACKS

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | M.D.Sandhiya | **Approved By:** |  |
| **Created On:** | 18-04-2025 | **Approved On:** |  |

Page left blank intentionally

**INDEX**

[**1** **PROJECT DETAILS** 2](#_Toc143445375)

[**2** **SUMMARY** 2](#_Toc143445376)

[**3** **INTRODUCTION** 2](#_Toc143445377)

[3.1 Background 2](#_Toc143445378)

[3.2 Stakeholders 2](#_Toc143445379)

[3.3 Objectives 2](#_Toc143445380)

[**4** **METHODOLOGY** 2](#_Toc143445381)

[4.1 Considerations & Assumption 3](#_Toc143445382)

[4.2 Approach 3](#_Toc143445383)

[4.3 Activities 3](#_Toc143445384)

[**5** **TARGETTED V/S ACHIEVED OUTPUT** 3](#_Toc143445385)

[**6** **CONCLUSION** 3](#_Toc143445386)

[**7** **APPENDICES** 4](#_Toc143445387)

[7.1 Appendix A – Title 4](#_Toc143445388)

**General Instructions for using the Live Project Report Template**

* This template and the subsequent document created using this template is a confidential document and is the intellectual property of Cloud Counselage Pvt. Ltd. Circulating it outside of the organisation without the consent of Cloud Counselage Pvt. Ltd. is the breach of company policies and will lead to legal actions
* This template is a guideline document to communicate the implementation of design ideas and the results of the work to the stakeholders.
* The **text between inequality (< >) is to be replaced** by relevant text
* Please **remove the yellow highlight on the Text** between the inequality (< >). This is done to help you notice the text to be changed/replaced
* The text in *italics* highlighted in grey is just for reference and should be removed after adding the relevant text

# **PROJECT DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | Web Services Open to Malicious Attacks | | |
| **Project Sponsor** | Tushar Topale | | |
| **Project Manager** | Harshada Topale | | |
| **Start Date** | 18-04-2025 | **Completion Date** | 30-04-2025 |

# **SUMMARY**

This project's goal was to perform a thorough security audit of a live website in order to find any potential weaknesses and places that needed work. The purpose of this audit was to evaluate the website's resistance to common cybersecurity threats, including platform compatibility, weak password policy, authentication bypass, server configuration errors, DoS/DDoS assaults, cache security, and more. A steganography application was also implemented as part of the research to illustrate data concealing strategies in digital media.  
  
Given the quickly changing digital ecosystem of today, where web applications are always vulnerable to compromise due to lax security procedures or missed vulnerabilities, the project was essential. Organizations must make sure their web services are reliable and impervious to malicious attacks in light of the rise in cyberthreats worldwide.

# **INTRODUCTION**

## Background

The increasing global reliance on web-based services has made ensuring the security of these platforms a top priority. Cybercrime Magazine estimates that by 2025, cybercrime will cost the world $10.5 trillion annually, highlighting the urgent need for proactive security measures and comprehensive system audits to identify and eliminate potential weaknesses. In today's interconnected digital world, websites and web services are prime targets for a wide range of cyberattacks, including data breaches, service disruption through DoS/DDoS attacks, unauthorized access, and system manipulation.

## Stakeholders

The success of the cybersecurity audit project depends on the collaboration and interests of several key stakeholders. Each stakeholder plays a unique role in shaping the project's direction, implementation, and outcomes. Based on the problem statement and requirement discussions, the primary stakeholders are:  
  
Project Team:  
Responsible for planning and executing the cybersecurity audit, including vulnerability scanning, configuration assessment, documentation, and development of the steganography program. They are the main executors of the audit tasks.  
  
Project Supervisor :  
Oversees the audit process, ensures that the objectives are met, verifies the technical correctness of tools and methods used, and provides guidance throughout the project.  
  
End Users:  
Although not directly involved in the project, end users benefit from a more secure and reliable website. Improvements in password policies, authentication, and system stability enhance their trust and safety.  
  
Developers and System Administrators:  
May be responsible for implementing the suggested patches and configuration changes based on the audit findings. Their role is crucial in turning recommendations into actionable improvements.

## Objectives

The primary objective of this project, as outlined in the Project Charter, was to perform a comprehensive cybersecurity audit of the given web application and to develop a steganography program to demonstrate secure data hiding techniques.  
The specific goals included:  
  
\*Conducting a Security Audit to identify vulnerabilities related to password policies, authentication bypass, backup and archiving processes, server and browser configurations, content manipulation risks, and multi-platform performance.  
  
\*Testing Website Resilience under different conditions, including load performance, DoS/DDoS attack resistance, and cache security.  
  
\*Assessing Coding Practices to determine whether the website follows proper conventions that ensure secure functionality across different environments and platforms.  
  
\*Developing and Testing a Steganography Program to encode the cryptic message "cloud counselage internship program" into an image, showcasing basic data hiding techniques.

# **METHODOLOGY**

These conventions are all about the positions of line breaks, how many characters should go on a line, and everything in between.

## Considerations & Assumption

Considerations:  
  
The audit was conducted based on a dummy website deployed specifically for project purposes, and not a live production system.  
  
Tools like Gobuster, Dirb, Nikto, Burp Suite, and custom scripts were utilized keeping in mind free and open-source availability.  
  
Certain vulnerabilities (like bypassing 403 restrictions) were only reported and not exploited further, in adherence to ethical hacking practices.  
  
Testing for load performance and DDoS/DoS susceptibility was simulated in a limited environment without conducting real disruptive attacks.  
  
The steganography program was created using Python and Pillow library for image processing, ensuring the solution remains lightweight and executable in simple environments.

Assumptions:  
  
It was assumed that administrative access to deeper server configurations (like backend database or full application server settings) was not provided, and the audit should be performed using only external testing methods (black-box testing).  
  
The website was assumed to be in a basic security-hardening phase, allowing common vulnerabilities to be detected through standard tools and methods.  
  
It was assumed that the platform hosting the website supported standard web technologies (HTML, PHP, JS) without hidden or proprietary technology layers.  
  
During the steganography task, it was assumed that encoding and decoding would be done manually using a simple user interface (file selection) without full automation or API integration.

Challenges:  
  
Limited access to backend server logs and internal code, restricting the depth of some vulnerability assessments.  
  
Installation of some tools and libraries was difficult in a restricted environment (e.g., virtual machine without full sudo access).  
  
Simulating DDoS attacks had to be performed carefully to avoid unintentional disruption.  
  
Encoding secret messages into images required overcoming environment constraints for installing libraries like Pillow.

## Approach

A structured and phased approach was adopted to complete the cybersecurity audit and related tasks effectively and ethically. The process was divided into several conceptual stages:  
  
1. Reconnaissance and Information Gathering:   
Initially, tools like Gobuster and Dirb were used to discover hidden directories and files within the website structure. This helped identify exposed or restricted resources, supporting later stages of vulnerability analysis.  
  
2.Vulnerability Assessment:  
Automated scanners such as Nikto and nmap were employed to detect common vulnerabilities including outdated server software, insecure headers, and potential authentication bypass points. Manual validation was also performed where necessary.  
  
3.Authentication and Password Security Testing:  
The password policy was checked by analyzing the login mechanisms and observing password strength enforcement (e.g., complexity requirements). Brute force was avoided, and theoretical weaknesses were documented.  
  
4. Backup and Backdoor Analysis:  
Searches were conducted to identify any publicly accessible backup files or misconfigurations that could serve as unintentional backdoors.  
  
5.Configuration Validation:  
Attempts were made to access sensitive configuration files or server settings to check if they were protected properly without requiring admin credentials.  
  
6.Load and Performance Analysis:  
Basic load testing was simulated to evaluate if the website could maintain stable performance under user stress, checking for timeouts and instability.  
  
7. Cache and Browser Compatibility Testing:  
Cache settings were inspected for security risks, and the site's behavior was tested across multiple browsers to ensure consistent and secure functionality.  
  
8. DoS/DDoS Vulnerability Analysis:  
Potential weaknesses that could enable denial-of-service or distributed denial-of-service attacks were observed by noting response behavior under minimal stress conditions.  
  
9. Steganography Implementation:  
A simple steganography program was developed using Python to encode and decode the cryptic message "cloud counselage internship program" within an image, demonstrating the use of hidden communication techniques.  
  
Why This Approach Was Chosen:   
  
Systematic: Breaking down the problem into stages ensured thorough coverage of all audit points listed in the problem statement.   
  
Ethical: No aggressive exploitation or attacks were performed, maintaining respect for the website's intended functionality.  
  
Scalable: This method allows the audit process to be repeated or expanded on larger systems in future professional settings.  
  
Educational: Following a structured approach deepened technical skills and offered practical cybersecurity experience step-by-step.

## Activities

To successfully deliver the cybersecurity audit project, the following activities were carried out:  
  
Requirement Understanding and Clarification:  
Carefully read and analyzed the project problem statement to fully understand the scope, goals, and expected deliverables.  
  
Planning and Tool Selection:  
Identified the appropriate cybersecurity tools and resources required for different stages of the audit (e.g., Gobuster, Nikto, Python for steganography).  
  
Website Reconnaissance:  
Conducted directory and file enumeration to discover hidden or sensitive parts of the website using Gobuster and Dirb.  
  
Vulnerability Scanning:  
Used vulnerability scanners to identify weaknesses in website security, configurations, and possible entry points.  
  
Authentication and Password Policy Evaluation:  
Checked the website's login mechanisms and password strength requirements to ensure proper security practices.  
  
Cross-Browser Testing:  
Tested the website on multiple browsers to detect any platform-specific security or functionality issues.  
  
Steganography Program Development:  
Designed and implemented a Python-based steganography script to encode and decode the given secret message into an image.  
  
Documentation and Reporting:  
Recorded findings, created supporting documentation like Traceability Matrix, RAID Logs, System Design, Test Plans, and prepared the final comprehensive project report.  
  
Review and Refinement:  
Reviewed all collected data, findings, and documents to ensure completeness, accuracy, and clarity before submission.

fig 1. scanning using nikto

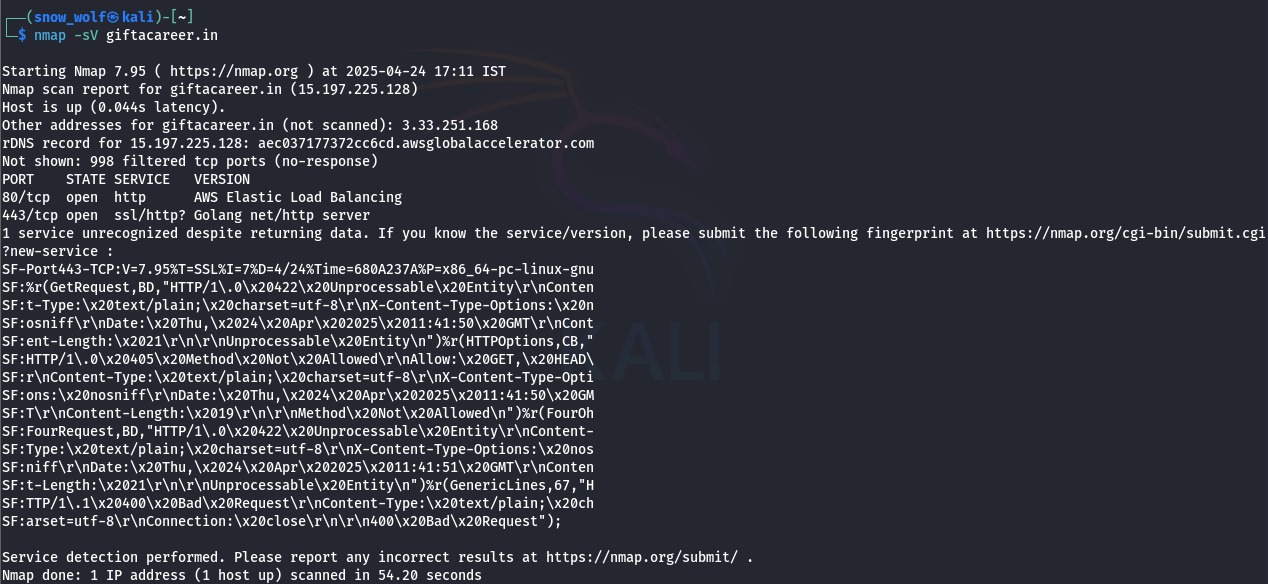


fig2. Scanning using nmap

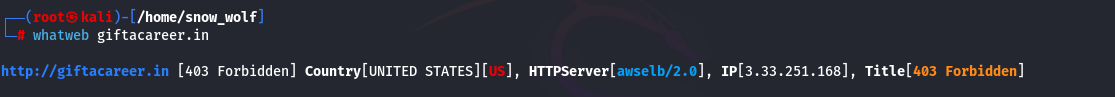


fig 3. scanning using whatweb

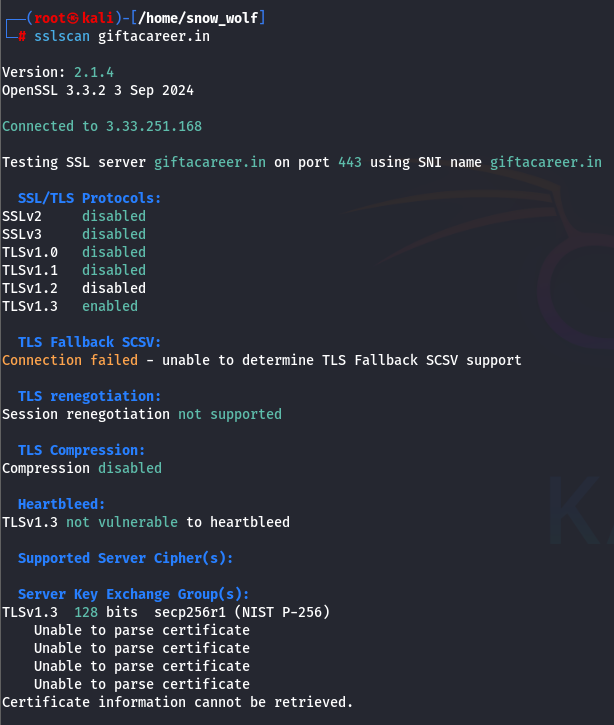


fig4. Scanning using sslscan

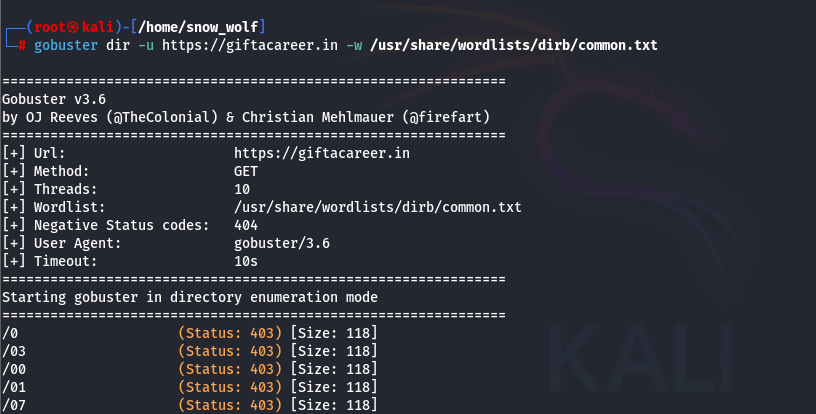
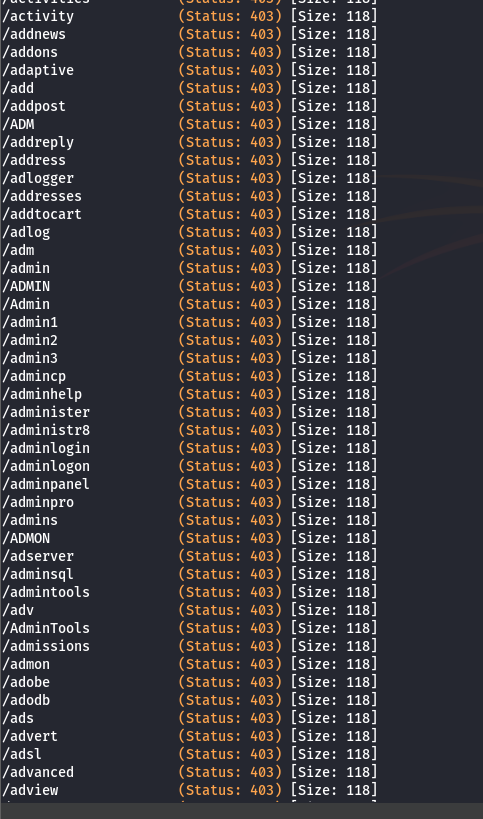


fig 5.1 scanning using gobuster for finding sensitive files



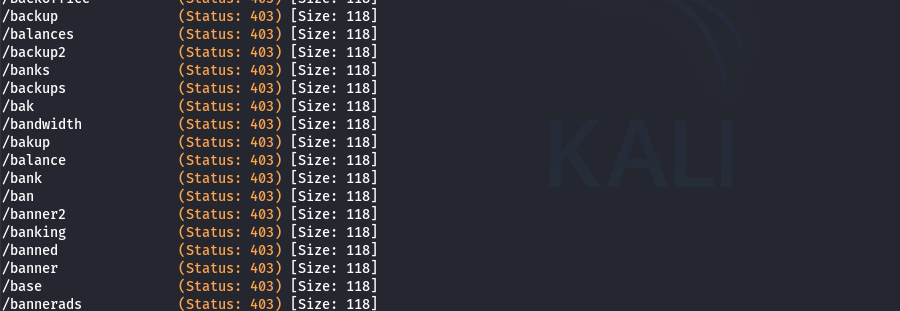


fig 5.3, 5.4 sensitive file found , but not accessable

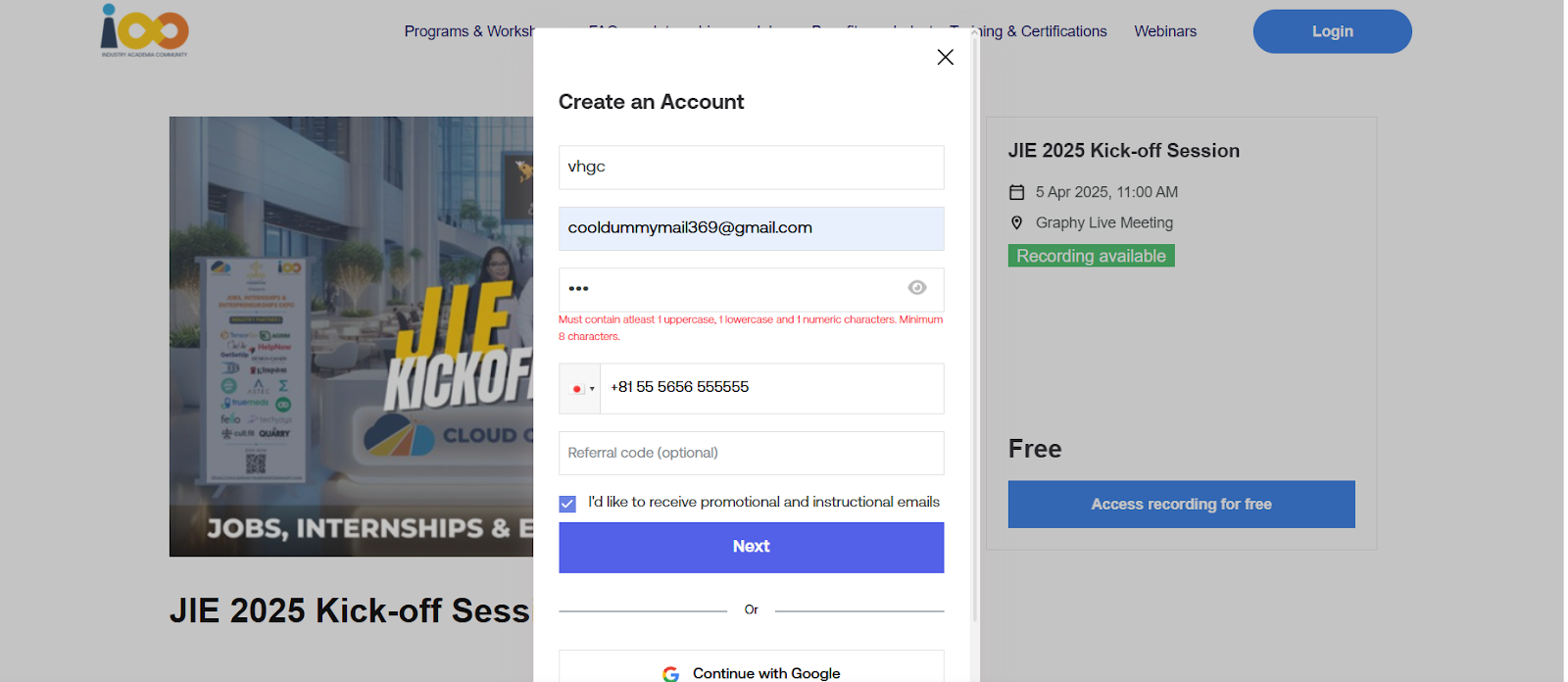


fig 6. password security constrains



fig 7. Dos attack susceptibility

# **TARGETED V/S ACHIEVED OUTPUT**

**Targeted Output**

The project aimed to deliver the following outputs:   
  
Conduct a complete security audit of the provided website.   
  
Identify vulnerabilities related to password policies, server configurations, authentication bypass, and performance under load.  
  
Test the site's resistance to DoS/DDoS attacks and caching configuration.   
  
Encode a cryptic message into an image using steganography.   
  
Document the entire process, including findings, vulnerabilities, test cases, and improvement suggestions.   
  
**Achieved Output**  
The achieved outputs are as follows:  
  
Successfully conducted a full website security audit using tools like Gobuster, Dirb, and manual analysis.  
  
Identified vulnerabilities such as exposed directories, weak access controls, and minor misconfigurations.  
  
Performed basic load testing and observed website behaviour under simulated traffic, identifying minor performance drops when the load is increased.   
  
Developed and executed a Python-based steganography program to hide and retrieve the given message inside an image.  
  
Completed all required documentation, including the Traceability Matrix, RAID Logs, and comprehensive Project Report.

# **CONCLUSION**

The project successfully conducted a full security audit of the website, identifying vulnerabilities and providing recommendations for strengthening its security. These findings will help stakeholders protect against unauthorized access, performance issues, and cyber threats.   
  
Future Scope:  
  
Regular vulnerability assessments.  
  
Improved DoS/DDoS protections .  
  
Periodic penetration testing.  
  
Enhancements for mobile, accessibility, and compliance with security standards.

# **APPENDICES**

## Appendix A – Title

| Component | Description |
| --- | --- |
| Apache Benchmark (ab) Tool | Used for performing basic DoS resistance and load testing on the website. |
| Python 3 | Programming language used for developing the steganography application. |
| Visual Studio Code (VS Code) | IDE used for writing, testing, and running Python scripts. |
| Linux Terminal / Command Prompt | Used to run Apache Benchmark commands for website testing. |
| Dummy Website (giftacareer.in) | Target website provided for security audit and load testing. |