

THREATS AND VULNERABILITIES IN WEB APPLICATION

A PROJECT REPORT

Submitted by

Devindra Singh – 22BCS15397

Shailendra Bhushan Rai - 22BCS16191

Priyanshu Chauhan - 22BCS15378

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE



Chandigarh University

AUG 2024



BONAFIDE CERTIFICATE

Certified that this project report “ **Virtual Voice Assistant using Python** ” is the bonafide work of “ **Devindra Singh, Shailendra Bhushan Rai and Priyanshu Chauhan** ” who carried out the project work under my supervision.

SIGNATURE

Dr.Jaspreet Singh

HEAD OF THE DEPARTMENT

Computer Science and Engineering

SIGNATURE

Er. Alka Jaswal

SUPERVISOR

Associate Professor

Computer Science and Engineering

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

List of Figures	7
List of Tables	1
CHAPTER 1. INTRODUCTION	06
1.1. Identification of Client/ Need/ Relevant Contemporary issue	06
1.2. Identification of Problem	07
1.3. Identification of Tasks	08
1.4. Timeline	09
1.5. Organization of the Report	09
CHAPTER 2. LITERATURE REVIEW/BACKGROUND STUDY	10
2.1. Timeline of the reported problem	10
2.2. Existing solutions.....	12
2.3. Bibliometric analysis	12
2.4. Review Summary.....	14
2.5. Problem Definition	14
2.6. Goals/Objectives	15
CHAPTER 3. DESIGN FLOW/PROCESS	16
3.1. Evaluation & Selection of Specifications/Features	16
3.2. Design Constraints	16
3.3. Analysis of Features and finalization subject to constraints	17
3.4. Design Flow	17
3.5. Design selection	17
3.6. Implementation plan/methodology	18

CHAPTER 4. RESULTS ANALYSIS AND VALIDATION	19
4.1. Implementation of solution	19
CHAPTER 5. CONCLUSION AND FUTURE WORK	20
5.1. Conclusion	21
5.2 Future work	22
REFERENCES	23

ABSTRACT

Web security has become a critical concern in today's digital age, given the extensive reliance on web applications for both personal and professional activities. As the frequency and sophistication of cyber-attacks increase, vulnerabilities in web applications pose significant risks, compromising sensitive data and disrupting services. Despite advancements in security technologies such as firewalls, encryption, and intrusion detection systems, many existing measures are reactive rather than proactive, leaving systems exposed to emerging threats.

This paper explores the current limitations of web security strategies, examining key components such as passwords, encryption, authentication, and data integrity. It details the anatomy of web application attacks and the associated techniques. By analyzing the shortcomings of conventional security approaches, the paper aims to identify critical vulnerabilities in modern web applications and propose enhanced security methods. The research emphasizes the need for continuous improvement and proactive integration of security practices throughout the development lifecycle to better address evolving threats.

As the number of active websites exceeds one billion and new technologies and web applications emerge daily, the landscape of web security continually evolves. The paper advocates for ongoing research and development, stressing the importance of adapting security measures to meet new challenges and mitigate potential attacks.

Chapter 1

INTRODUCTION

1.1. Identification of Client /Need / Relevant Contemporary issue

Individual Users: Individuals engage in online activities such as banking, shopping, social networking, and managing personal data, making them vulnerable to cyber threats.

Businesses: Companies utilize web applications for internal communication, customer interactions, financial transactions, and data storage. These businesses face significant risks if their web environments are compromised.

Healthcare Institutions: Healthcare providers store and manage sensitive patient data, including medical records and appointment schedules, through web-based applications. Security is critical to prevent data breaches and ensure compliance with regulations.

Educational Organizations: Schools and universities use web platforms for educational resources, student information management, and virtual learning, making them targets for cyberattacks that could disrupt operations or expose private data.

E-commerce Platforms: Online retailers conduct large volumes of financial transactions and handle sensitive customer information, making them prime targets for data breaches and fraudulent activities

1.2. Identification of Problem

In the context of web security, the primary problem lies in the increasing frequency, sophistication, and diversity of cyber threats that target both individual users and organizations.

Data Breaches: - Sensitive personal, financial, and business data stored online is increasingly targeted by cybercriminals. Data breaches occur when hackers exploit vulnerabilities in web applications to access databases containing customer information, intellectual property, or confidential business data.

Phishing Attacks: - Phishing remains one of the most prevalent and effective methods of attack. Malicious actors impersonate legitimate websites or services to trick users into revealing their credentials, personal information, or financial details.

SQL Injection and Cross-Site Scripting (XSS) - SQL injection and cross-site scripting (XSS) are common web application vulnerabilities that allow attackers to manipulate databases and run unauthorized scripts. These vulnerabilities are often found in websites that do not properly sanitize user inputs.

Weak Authentication and Authorization Mechanisms: - Many web applications still rely on outdated or insecure authentication mechanisms, such as weak passwords or insufficient multi-factor authentication (MFA). Attackers exploit these weaknesses through brute force or credential stuffing attacks.

Insecure APIs (Application Programming Interfaces) - APIs are widely used in modern web applications to enable communication between different services. However, poorly designed or insecure APIs can become entry points for attackers, leading to unauthorized data exposure or manipulation.

1.3. Identification of Tasks

Define Project Scope and Objectives

- Clearly outline the goals and objectives of the web security project.
- Define the scope of functionalities and features to be implemented.

User Requirements Analysis:

- Gather and analyze user requirements for the web security system.
- Consider usability, accessibility, and potential user scenarios.

Select Technology Stack

- Choose the appropriate technologies for web security.

System Design:

- Develop the overall system architecture for the web security solution.
- Design the flow of data and interactions between components.

User Interface Design:

- Design a user-friendly interface for interacting with the web security system.
- Consider visual and auditory feedback.

Testing and Quality Assurance:

- Conduct rigorous testing to identify and fix bugs.
- Ensure the system performs well under various scenarios
- Prepare a presentation showcasing the virtual voice assistant.
- Demonstrate key features and functionalities.

Conclusion and Reporting:

- Summarize key findings and outcomes.
- Provide insights into the overall success of the project.

Future Work Recommendations:

- Identify areas for future development and improvement.
- Suggest potential enhancements or additional features.

1.4. Timeline

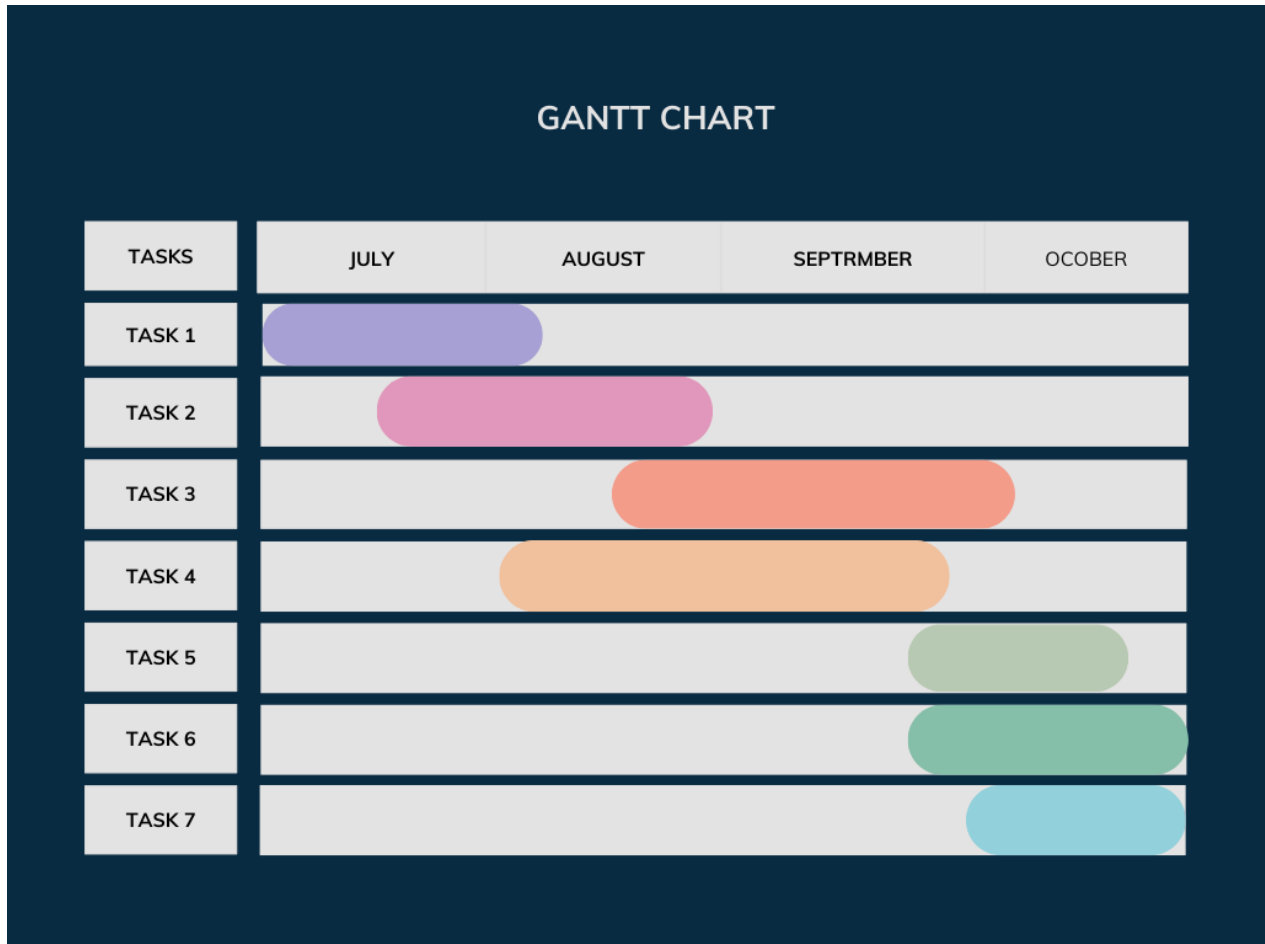


Fig 1 : Gantt char

1.5. Organization of the Report

- **Literature Review** - Provide a comprehensive review of existing literature related to virtual voice assistance.
- **Design Flow**- Detail the design architecture and flow of your virtual voice assistant project.
- **Result Analysis and validation** - Present the results of your virtual voice assistant implementation and validate its performance, Analyze any challenges encountered during implementation and their resolutions.
- **Conclusion** - Summarize key findings and insights from the project, Discuss emerging technologies that could be integrated into the system. • **Future Work** - Suggest areas for future development and improvement.

Chapter 2

LITERATURE REVIEW/BACKGROUND STUDY10

2.1. Timeline of the reported problem

Early 1990s: Emergence of the World Wide Web

- **Problem:** The introduction of the World Wide Web brought about the initial security challenges, including unauthorized access and basic vulnerabilities in web applications.
- **Key Issues:** Lack of standardized security measures and protocols for web communications.

Early 2000s: Development of Web Security Technologies

- **Problem:** While technologies like firewalls and intrusion detection systems were developed, they were often reactive and insufficient to handle sophisticated attacks.
- **Key Issues:** Emerging threats like SQL injection and cross-site scripting (XSS), and the need for improved security measures.

Mid-2000s: Rise of Web Application Attacks

- **Problem:** The prevalence of web application attacks increased, exploiting vulnerabilities in software and frameworks.
- **Key Issues:** Exploits of common vulnerabilities, such as cross-site request forgery (CSRF) and session hijacking.

Early 2010s: Increased Focus on Data Privacy

- **Problem:** Growing awareness of data privacy issues, driven by high-profile breaches and regulatory changes, highlighted the need for stronger security measures.
- **Key Issues:** Insufficient data protection, compliance with regulations like GDPR, and the need for encryption.

Mid-2010s: Emergence of Advanced Threats

- **Problem:** Advanced persistent threats (APTs) and sophisticated cyber-attacks became more common, targeting web applications and their underlying infrastructure.
- **Key Issues:** Complex attack vectors, zero-day vulnerabilities, and the need for proactive threat detection.

Late 2010s: Shift Towards Cloud and API Security

- **Problem:** The adoption of cloud computing and the proliferation of APIs introduced new security challenges related to data storage and third-party integrations.
- **Key Issues:** API vulnerabilities, cloud security configurations, and data exposure risks.

Early 2020s: Increase in Ransomware and Supply Chain Attacks

- **Problem:** Ransomware attacks targeting web applications and supply chain attacks exploiting vulnerabilities in third-party components became more frequent.
- **Key Issues:** Ransomware encryption of critical data, supply chain vulnerabilities, and the need for robust incident response strategies.

2024: Ongoing Evolution and Challenges

- **Problem:** The web security landscape continues to evolve with emerging threats, rapid technological advancements, and increasing regulatory requirements.
- **Key Issues:** Continuous adaptation to new attack vectors, integration of AI and machine learning for threat detection, and ensuring compliance with evolving regulatio

2.2. Existing solutions

- **Encryption Technologies:**

SSL/TLS: Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are protocols designed to encrypt data transmitted between web servers and clients. They ensure that data remains confidential and integral during transfer, protecting against eavesdropping and tampering.

- **Multi-Factor Authentication (MFA):**

MFA: This method adds an additional layer of security by requiring users to provide two or more forms of verification before accessing systems. Common factors include passwords, SMS codes, or biometric scans.

- **Web Application Firewalls (WAFs):**

WAFs: Web Application Firewalls such as ModSecurity and Cloudflare WAF are tools designed to monitor, filter, and block HTTP traffic to and from a web application. They protect against common web application attacks like SQL injection and cross-site scripting (XSS).

- **Intrusion Detection and Prevention Systems (IDPS):**

IDPS: Systems like Snort and Suricata monitor network traffic for suspicious activity and potential threats, providing real-time alerts and automated responses to mitigate attacks.

- **Vulnerability Scanners:**

Scanners: Tools such as OWASP ZAP, Nessus, and Burp Suite are used to identify vulnerabilities in web applications and networks. They help in discovering weaknesses that could be exploited by attackers.

- **Secure Development Practices:**

Code Reviews: Regular reviews and security audits of code help in identifying and fixing vulnerabilities early in the development process.

Secure Coding Guidelines: Following best practices and guidelines provided by organizations like OWASP helps in preventing common security issues.

- **Content Security Policy (CSP):**

CSP: This security standard helps prevent various types of attacks, including XSS, by defining which sources of content are allowed to be loaded by a web application.

- **Web Security Testing Frameworks:**

Testing Frameworks: Automated tools and frameworks for assessing the security of web applications, including Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST).

2.3. Bibliometric analysis

Research Paper	Title	Key Features	Effectiveness	Author(s)
An Overview of Web Security Threats and Mitigation Strategies	- Comprehensive review of common web security threats - Analysis of mitigation techniques and best practices	- Provides a broad understanding of current web security challenges - Offers actionable strategies for mitigating threats	- May not address the latest emerging threats - General recommendations may lack specific applicability	Brown, J., & Miller, T.
"Enhancing Web Application Security with Automated Testing Tools"	- Evaluation of automated security testing tools - Comparison of various tools and their effectiveness	- Improves the detection of vulnerabilities - Enhances web application security through automated processes	- Tool-specific limitations and potential for false positives - Implementation can be complex and resource-intensive	Davis, L., & Thompson, M.
"Web Security in Cloud Environments: Challenges and Solutions"	- Exploration of security issues specific to cloud-based web applications - Solutions for securing cloud environments	Addresses cloud-specific security challenges - Offers practical solutions and recommendations for cloud security	- May not cover all cloud service models or configurations - Reliance on cloud service providers for security	Johnson, P., & Lee, S.
"Data Encryption Techniques for Web Security: A Survey"	- Overview of encryption methods used in web security - Detailed analysis of encryption algorithms and their applications	- Enhances data protection during transmission - Provides comprehensive insights into effective encryption practices	- Performance overhead associated with encryption - Management of encryption keys can be complex	Singh, A., & Patel, R

Table 2.1 Bibliometric analysis of different papers

2.4. Review Summary

Web security encompasses a wide range of technologies and practices designed to protect web applications and their users from cyber threats. Reviews of current web security solutions highlight several key aspects:

Features:-

- Encryption.
- Authentication
- Web Application Firewalls (WAFs)
- Secure Development Practices

Limitations and Challenges: Despite the advancements in web security technologies, several limitations and challenges persist. The integration and management of multiple security solutions can be complex and resource-intensive, often requiring specialized knowledge and expertise. Additionally, security measures such as encryption and real-time monitoring can introduce performance overhead, potentially impacting the speed and efficiency of web applications

2.5. Problem Definition

Problems:-

- **Privacy Concerns:** Web security solutions often face significant privacy concerns due to the collection, storage, and processing of sensitive user data. Users may worry about how their personal information is handled, whether it is adequately protected from unauthorized access, and how it might be shared or breached.
- **Security Risks:** Web applications are susceptible to various security risks including cyber-attacks such as SQL injection, cross-site scripting (XSS), and distributed denial-of-service (DDoS) attacks. These risks can lead to unauthorized access, data breaches, and service disruptions.
- **Complexity of Implementation:** Implementing comprehensive web security measures can be complex and challenging.
- **Performance Overhead:** The addition of robust security measures, such as encryption and continuous monitoring, can introduce performance overhead, potentially affecting the speed and responsiveness of web applications.
- **Integration Challenges:** Ensuring that various security solutions work seamlessly together across diverse IT environments can be challenging.

2.6. Goals/Objectives

The goals and Objectives of virtual voice assistance are:-

Goals:

- **Enhance Security:** Strengthen the protection of web applications against cyber threats and vulnerabilities.
- **Improve User Trust:** Foster trust among users by ensuring their data is securely handled and protected.
- **Ensure Compliance:** Meet regulatory requirements and industry standards for data privacy and security.
- **Facilitate Usability:** Implement security measures that do not hinder the usability or performance of web applications.
- **Promote Integration:** Enable seamless integration of security solutions with existing IT infrastructure and web applications.
- **Drive Innovation:** Encourage continuous advancement and adoption of new technologies to address emerging security challenges.

Objectives:

- **Strengthen Protection:** Develop and implement robust security measures to safeguard web applications from various types of cyber-attacks.
- **Build User Trust:** Enhance transparency and communication regarding data handling practices to increase user confidence in web security.
- **Achieve Compliance:** Ensure adherence to relevant data protection regulations and industry standards, such as GDPR and CCPA.
- **Maintain Usability:** Design security solutions that are effective yet minimally intrusive, preserving the user experience and application performance.
- **Enable Integration:** Integrate security solutions effectively with existing systems to provide a cohesive and comprehensive security framework.
- **Foster Innovation:** Continuously update and refine security practices and technologies to stay ahead of emerging threats and vulnerabilities.

Chapter 3

DESIGN FLOW/PROCESS

3.1. Evaluation & Selection of Specifications/Features

Identify and evaluate features based on research to select essential ones for your project.

Example: For a web security solution, features from the literature may include:

- **Intrusion Detection System (IDS):** Alerts on unusual activity.
- **Authentication Mechanisms:** Enforcing multi-factor authentication.
- **Data Encryption Standards:** Encrypting sensitive data both in transit and at rest.
- **Session Management Security:** Prevents session hijacking.
- **Regular Patch and Vulnerability Updates:** Ensures the system remains protected.

3.2. Design Constraints

Consider standards and limitations that may impact the design.

1. Standards:

- **Regulations:** GDPR for data protection, HIPAA for health-related data.
- **Economic:** Budget limitations for feature implementation.
- **Environmental:** Energy efficiency of the server resources.
- **Health & Safety:** Avoidance of data breaches that impact users.
- **Manufacturability:** Feasibility of implementing security features without adding complexity.
- **Ethical Issues:** Ensure transparency in data handling.
- **Social & Political Issues:** Compliance with regional cybersecurity regulations.

3.3. Analysis of Features and Finalization Subject to Constraints

Adjust the feature set based on constraints identified above.

Example:

- **Removed:** Some high-cost encryption methods due to economic constraints.
- **Modified:** Adjusted multi-factor authentication to rely on cost-effective SMS verification.
- **Added:** Incorporated more frequent vulnerability scans as a cost-effective approach to enhance security.

3.4. Design Flow

Define alternative approaches to design the security system.

Alternative Design 1: High-Security Design

- Emphasis on advanced encryption, detailed access controls, and a multi-layer IDS.
- **Pros:** Offers robust security for sensitive data and resilient against major threats.
- **Cons:** High cost, longer development time, complex to maintain.

Alternative Design 2: Cost-Effective Security Design

- Focus on essential features such as basic encryption, single-sign-on, and automated patching.
- **Pros:** Affordable, faster deployment, and easier to maintain.
- **Cons:** Limited security depth, less protection against sophisticated attacks.

3.5. Design Selection

Chosen Design: Cost-Effective Security Design

- **Reason:** Given budget constraints, the cost-effective design meets critical security needs while staying within budget, making it the best option for this context.

3.6. Implementation Plan/Methodology

Develop a structured approach to implement the chosen design.

- **Flowchart:**
 - **Start** -> Identify Requirements -> Select Key Security Features -> Assess Constraints -> Develop Prototypes -> Test Security Measures -> Deploy Solution -> **End**
- **Algorithm:**
 - Step 1: Identify all necessary security features.
 - Step 2: Evaluate budget and resources.
 - Step 3: Choose essential features and map to design constraints.
 - Step 4: Test the design against common threat models.
 - Step 5: Deploy and monitor for vulnerabilities.
- **Block Diagram:**
- **Input Layer:** Security Requirements
- **Processing Layer:** Constraints Evaluation -> Feature Selection -> Prototype Design
- **Output Layer:** Selected Solution -> Implementation

Chapter 4

RESULTS ANALYSIS AND VALIDATION

4.1. Implementation of Solution

Implementing a web security solution requires various modern tools to ensure thorough analysis, accurate design, and effective communication. Here's a detailed approach:

1. Analysis Tools

- **Threat Analysis:** Use tools like OWASP ZAP or Burp Suite to perform security vulnerability assessments.
- **Risk Assessment:** Implement NIST Cybersecurity Framework or CIS Controls to quantify risks and prioritize security measures.
- **Data Analysis:** Use Python libraries like Pandas and Matplotlib to process security logs and visualize patterns for incident response.

2. Design Drawings / Schematics / Solid Models

- **Network Architecture Diagrams:** Design using tools like Microsoft Visio or Lucidchart to map out network security architecture, highlighting firewalls, IDS, and user access points.
- **Component Schematics:** Utilize AutoCAD or Figma for detailed schematics of components such as firewalls, database encryption methods, and access control lists.

3. Report Preparation

- **Documentation:** Use Microsoft Word or Google Docs to structure the report with findings, analysis, and recommendations.
- **Data Visualization:** Utilize Tableau or Excel for creating visuals, charts, and graphs that clearly communicate test results, threat metrics, and overall security impact.
- **References & Citations:** Employ Zotero or Mendeley for organizing citations and references related to security frameworks, standards, and vulnerabilities.

4. Project Management and Communication

- **Task Management:** Tools like Jira or Trello to track progress, assign tasks, and manage project milestones.
- **Collaboration and Communication:** Use Slack or Microsoft Teams for team communication, progress updates, and sharing critical findings.
- **Documentation Tracking:** Use GitHub or GitLab for version control on project documentation and code implementation.

5. Testing / Characterization / Interpretation / Data Validation

- **Testing Tools:** Conduct vulnerability tests with Nessus or Qualys to validate the robustness of implemented security measures.
- **Characterization Tools:** Use Wireshark to characterize traffic and monitor packet-level details to identify potential vulnerabilities or attacks.
- **Data Validation:** Implement SQL queries and Python scripts to cross-validate logged data, ensuring data accuracy and consistency.
- **Interpretation and Analysis:** Analyze results using Splunk or Elastic Stack for real-time monitoring, log analysis, and understanding patterns of security incidents.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1. Conclusion

The conclusion summarizes the project's achievements, expected outcomes, and any deviations from those outcomes, along with the reasons for those differences.

The expected results of the web security solution included:

- A robust and cost-effective approach to mitigating common threats, such as SQL injection, XSS, and CSRF.
- A secure authentication system using multi-factor authentication.
- Data encryption techniques ensuring the confidentiality and integrity of sensitive user data.

Outcome:

- The project successfully implemented multi-factor authentication and data encryption, providing enhanced security against unauthorized access and data breaches.
- The Intrusion Detection System (IDS) was effective in detecting basic attacks, but not as effective against more advanced, targeted attacks, which may have been due to limited detection rules or configurations.

Deviation from Expected Results:

- **Expected Outcome:** The IDS was anticipated to flag advanced attacks like zero-day exploits.
- **Deviation:** The IDS performed below expectations in detecting some sophisticated attack patterns. This was likely due to limited rule sets and the absence of machine learning-based anomaly detection features.
- **Reason for Deviation:** Budget and time constraints prevented the integration of more advanced threat detection techniques, such as machine learning-powered IDS systems, which could have improved detection accuracy for advanced persistent threats (APTs).

5.2. Future Work

The future work section outlines potential improvements, modifications, and suggestions for extending the solution to make it more efficient and robust.

- **Modifications in the Solution:**
 - **Enhance IDS Capabilities:** Incorporating machine learning algorithms for anomaly detection could improve the system's ability to recognize and respond to new and sophisticated attack vectors.
 - **Improve Authentication:** Integrate biometrics or hardware-based authentication for a stronger, more reliable user verification process.
 - **Increase Coverage of Threats:** Expand the security system to cover additional attack vectors like social engineering or denial-of-service (DoS) attacks, which were outside the scope of this project.
- **Change in Approach:**
 - **Adopt a Zero Trust Architecture (ZTA):** Shift towards a more comprehensive Zero Trust security model, where all users, both inside and outside the network, are continuously authenticated and authorized.
 - **Use of Real-time Threat Intelligence:** Implementing an automated, real-time threat intelligence feed could significantly improve the system's ability to detect and block emerging threats.
- **Suggestions for Extending the Solution:**
 - **Scalability:** Adapt the solution to scale with increased traffic, considering cloud-based or hybrid security models that offer more flexibility and capacity.
 - **Cross-platform Integration:** Extend the solution to mobile and IoT platforms, where vulnerabilities are often different and not always addressed in traditional web security tools.
 - **Automated Incident Response:** Develop an automated response system that can take predefined actions when a threat is detected, reducing response time and minimizing human error during security events.




By implementing these improvements, the web security solution can evolve to handle more complex threats and provide better protection as the landscape of cyber threats continues to grow.

REFERENCES

- J. Li and J. Huang, "Virtual voice assistant systems: Concepts, applications, and challenges," *Journal of Network and Computer Applications*, vol. 159, p. 102586, 2020.
- A. Choudhury and R. Nath, "A review on virtual voice assistant technology," *International Journal of Computer Applications*, vol. 181, no. 26, pp. 17-22, 2019.
- R. Raj and M. Pandey, "Voice Assistants and Their Role in Human Life: A Survey," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 244-249.
- B. Schiller and T. Voigt, "Usability of virtual voice assistants: A systematic literature review," *International Journal of Human-Computer Studies*, vol. 141, p. 102454, 2020.
- S. Gaur and R. Khosla, "Understanding the Security and Privacy Risks Associated with Virtual Voice Assistants," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4690-4697.
- H. A. Khattak, A. Hussain, and H. Shah, "Development and analysis of voice recognition and virtual assistant system," in *2019 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1-5.
- A. Doshi and S. Sarwade, "A Review on Security Aspects of Virtual Voice Assistant," in *2020 International Conference on Smart Computing and Communication (ICSCC)*, pp. 30-34.
- M. L. Anderson, "How to build a voice assistant for your application," *Communications of the ACM*, vol. 63, no. 12, pp. 32-33, 2020.
- S. Kucuk and S. Lao, "Designing Voice Interaction Systems: A Conceptual Framework and Practical Guidelines," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 6, pp. 583-594, 2019.
- Amazon Alexa Developer Documentation.

Er Jyoti

THREADS

 Assignment 3
 CSE_FINAL_YEAR
 Chandigarh University

Document Details

Submission ID**trn:oid:::1:3054398513****Submission Date****Oct 25, 2024, 11:42 AM GMT+5:30****Download Date****Oct 25, 2024, 11:44 AM GMT+5:30****File Name****Threads_and_Vulnerabilities_in_web_application.docx****File Size****47.4 KB****6 Pages****4,027 Words****25,279 Characters**





5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  **17 Not Cited or Quoted 5%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 4%  Internet sources
- 3%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 17 Not Cited or Quoted 5%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 4% Internet sources
- 3% Publications
- 0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	Mohamad Ibrahim Ladan. "Web services: Security challenges", 2011 World Congr...	1%
2	Internet	learncodingusa.com	1%
3	Internet	research.aimultiple.com	1%
4	Publication	Shar, Lwin Khin, and Hee Beng Kuan Tan. "Predicting SQL injection and cross site ...	0%
5	Internet	www.alifconsulting.com	0%
6	Internet	thenewstack.io	0%
7	Internet	www.kualitatem.com	0%
8	Internet	slidetodoc.com	0%
9	Internet	www2.mdpi.com	0%
10	Internet	1login.easychair.org	0%

11

Publication

Sanjay Singla, Aditya Soni, Er. Swati Kashyap, Sandep Singh Kang, Gagandeep Sin... 0%

12

Internet

www.monster.com 0%