WebLAPS-AI Solution : Smart Analysis, Smarter Conversations

Presented by

- Sakshi Khalate – 25
- Layba Khan – 26
- Parshwa Mehta– 27
- Atharva Mohite– 28

Under the guidance
of
Dr.Jayasudha Koti

Department of Electronics and Telecommunication Engineering
St. Francis Institute of Technology, Mumbai

Contents

- Introduction and Motivation
- Problem Statement
- Objectives
- Flow Chart
- Literature Review
- Software Requirements
- Integration of AI and Cyber security
- Vulnerabilities That We Aim To Solve
- Requirement of Security in Applications
- References

Introduction

This project is about building a simple tool to help users find and fix security weaknesses (VULNERABILITIES) in web applications. Our tool has two parts:- A dashboard that gives a clear overview of all the security problems, and a friendly (INTERACTIVE) chatbot. When you want to fix a problem, the chatbot will tell you a simple story to explain what went wrong and give you easy step-by-step instructions to solve it.

Motivation

The main reason we are making this is because most security reports are very technical and hard to understand. This makes it difficult for users to fix security issues quickly and correctly, leaving their websites at risk. We want to make security easier and less scary for everyone, not just security experts. Our goal is to make fixing problems feel more like an interactive lesson, so users can solve today's issues and learn how to avoid them in the future.

Problem Statement

Users often struggle to fix security weaknesses in web applications because the reports from automated scanning tools are highly technical and difficult to understand. This communication gap leads to slow or incorrect fixes, which leaves applications vulnerable to attacks and creates a frustrating experience. The core problem is the absence of a tool that not only identifies these security flaws but also guides non-expert users through the repair process in an interactive, simple, and educational manner, failing to prevent similar mistakes from happening again in the future.

Objectives

- **Customizable Analytics:** Enable users to tailor dashboard views to their specific roles, responsibilities, and organizational requirements through conversational interfaces.
  For Example:-The dashboard can be tailored to different user roles. A developer might want to see a list of vulnerabilities in the code they wrote, while a project manager might want a high-level overview of the security posture of all their projects.

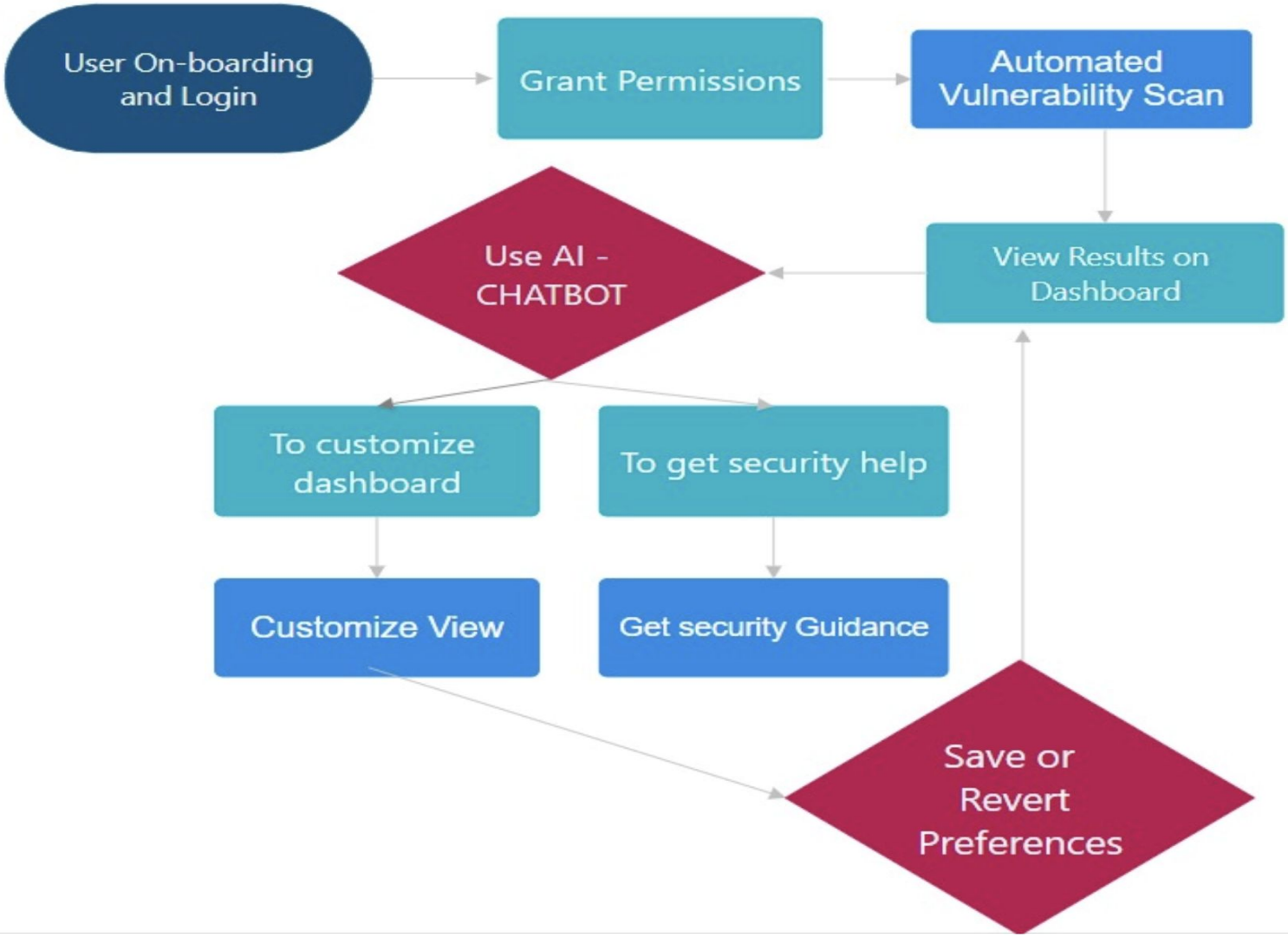
- **Proactive Security Culture:** Foster security awareness and best practices through engaging, story-based vulnerability explanations that transform technical information into actionable knowledge.
  For Example:-The chatbot can engage users in proactive security practices.
  Chatbot: "Congratulations on fixing the SQL Injection! Would you like a small summary about the same so that you can avoid it the next time.

- **Real-time Response:** Provide immediate threat detection and automated response.
  For Example:-An AI system can provide immediate threat detection and automated responses to minimize the window of opportunity for attackers.

Objectives

- **Enhanced Detection Accuracy:** An AI-powered system can use behavioral analysis to distinguish between normal user activity and malicious behavior.
  For example:- A user attempting to log in multiple times from various, geographically distant locations within a short time frame is a behavioral anomaly that could indicate a credential stuffing attack.
- **Intelligent Prioritization:** Develop risk-based scoring systems to focus attention on critical vulnerabilities first, enabling efficient resource allocation.
  For Example:-A project could develop a risk-based scoring system that assigns a score to each vulnerability, prioritizing the most critical ones. The score would be based on factors like:
  1.Severity
  2.Exploitability
  3.Business impact
- **Interactive Education:** Create storytelling chatbot capabilities to explain complex security concepts in accessible, narrative-driven formats that increase user engagement and retention.
  For Example:-The chatbot can use storytelling to explain complex concepts. Instead of just saying "You have an SQL Injection vulnerability," it can tell a story, which can be easy for human interaction.

Flow Chart


User On-boarding and Login
Grant Permissions
Automated Vulnerability Scan
View Results on Dashboard
Use AI - CHATBOT
To customize dashboard
To get security help
Customize View
Get security Guidance
Save or Revert Preferences

Literature Review

Sr. No.	Title of the Paper	Author	Descriptions	Results
1.	Identifying and Mitigating Web Application Vulnerabilities: A Comparative Study of Countermeasures and Tools [1]	Sadat, Sayed Elham & Naseri, Mohammed & Salamzada, Khosraw.	The paper analyzes the most critical web application security risks, focusing on the OWASP Top 10 vulnerabilities such as Broken Access Control, Injection flaws, Security Misconfigurations, and Cryptographic Failures. It evaluates their causes, impacts, and compares tools used to detect these threats. The study also discusses effective countermeasures, emphasizing continuous monitoring and proactive security to protect web applications.	The paper summarizes the top web application security risks, including Broken Access Control, Cryptographic Failures, Injection attacks, Insecure Design, and Security Misconfigurations. It evaluates the effectiveness of open-source tools in detecting these vulnerabilities and highlights best practices for mitigation. Continuous monitoring and adopting proactive security measures are emphasized to enhance web application resilience.
2.	Vulnerabilities and Security of Web Applications [2]	D. Yadav, D. Gupta, D. Singh, D. Kumar and U. Sharma	Web application vulnerabilities include common risks such as SQL injection, cross-site scripting (XSS), broken authentication, XML External Entities (XXE), security misconfigurations, and file upload flaws. These vulnerabilities allow attackers to access sensitive data, hijack sessions, or disrupt services. Effective protection requires input validation, secure coding, strong authentication, regular patching, and continuous monitoring.	Web application security is essential to protect sensitive data and ensure service availability. It involves defending applications against common threats like SQL injection, cross-site scripting, and broken authentication. Effective security requires secure coding, regular testing, strong authentication, and continuous monitoring to stay ahead of evolving cyber threats.

Literature Review

Sr. No.	Title of the Paper	Author	Descriptions	Results
3.	AI-Based Web Vulnerability Scanner: A Comprehensive Review [3]	Devadiga, Preeti and Varankar, Shruti and Kumari, Sneha and MISHRA, NILAMADHAB	The document reviews AI-powered web vulnerability scanners that improve accuracy by analyzing web traffic and code patterns to detect threats like SQL injection and XSS. It highlights a scorecard system that prioritizes vulnerabilities by risk, enabling efficient remediation. The AI approach reduces false positives and adapts to new threats, offering enhanced security over traditional methods.	The conclusion highlights that AI-enhanced vulnerability scanners significantly improve web security by increasing detection accuracy, reducing false positives, and prioritizing risks effectively through a scorecard system. This approach enables faster, focused remediation of critical threats and adapts continuously to emerging vulnerabilities. AI-driven scanners represent a major advancement over traditional methods, offering scalable and proactive web application protection for modern cyber threats.
4.	An Automatic Vulnerability Scanner for Web Applications with Firewall Techniques [4]	Rathod , S.K. Jagtap , J.R. Satpute , A. P. Shikhare , K.A. Pujari , A.S. Pandit	Key features of a web vulnerability scanner include automated asset discovery, comprehensive vulnerability detection, risk-based prioritization, and detailed actionable reports. Modern scanners offer continuous scanning, integration with development workflows, high accuracy with low false positives, and real-time alerts to efficiently manage and remediate security risks.	The paper concludes that combining automated vulnerability scanning tools with manual testing provides the most effective cybersecurity assessment. The developed scanner is flexible and customizable, capable of detecting various web and network threats. Its updating capability allows security researchers and auditors to address evolving vulnerabilities efficiently, making it a valuable asset for penetration testing and security audits.

Software Requirements

- Backend Framework: Python with **FastAPI** for high-performance API development and ML model integration.
- Machine Learning Libraries: **TensorFlow, PyTorch, scikit-learn** for vulnerability detection algorithms.
- Natural Language Processing: LangChain for orchestrating chatbot functionality , utilizing a powerful self-hosted LLM like **Llama 3** for generating story-driven explanations , and **spaCy** for efficient initial query parsing and entity extraction.
- Frontend Development: **React.js** for frame work , **Rechart** for visualization , **Redux Toolkit** for real time interactive components and threat detection , **Material UI of Figma** for ui components.
- Database Systems: **PostgreSQL** for relational data, **MongoDB** for vulnerability logs and user customizations.
- Scanning Engines: **OWASP ZAP** for web application vulnerability scanning, **Nikto** for rapid, initial scanning identifying known vulnerabilities and common misconfigurations on the web server itself , and custom build scanning models .
- Authentication: **JWT tokens** will be used to authenticate API requests , **OAuth 2.0** will serve as the framework for authorization and potential third-party logins.
- Development Environment: **VS code** for coding and **Git-Hub** for organising and saving code files in repositories

Integration of AI & Cyber Security

Integrating AI with cybersecurity provides:-

- **Spots Hidden Threats:** The AI constantly sifts through your network data to **find suspicious patterns** and **predict attacks** before they happen.

- **Acts Instantly to Stop Attacks:** The moment a threat is found, the **AI can automatically take action to block malicious traffic** or begin other security measures.

- **Focuses on What Matters Most:** It **intelligently prioritizes threats**, allowing you to focus your energy on the most critical security issues first.

- **Connects the Dots**: The AI **analyzes data from multiple sources** at once to give you a complete picture of a potential threat. In this project, the AI modules are designed to process real-time network traffic, flag suspicious activity, and even auto-mitigate threats.

Vulnerabilities That We Aim To Solve

- **XSS (Cross-Site Scripting) & CSRF (Cross-Site Request Forgery): XSS** allows attackers to inject malicious scripts into a website viewed by other users, potentially stealing data or hijacking sessions. **CSRF** tricks a logged-in user into performing unwanted actions on a web application (e.g., changing their email or transferring funds).
- **SQL Injection (SQLi):** Attackers inject malicious SQL code into input fields, allowing them to view, modify, or delete data in the application's database, leading to **data theft** or **loss of integrity.**
- **Buffer Overflow**: Occurs when a program tries to write more data to a fixed-length memory block (**buffer**) than it can hold. This can **corrupt adjacent data** or allow an attacker to **execute malicious code**.
- **Vulnerable & Outdated Components:** Using libraries, frameworks, or other software components with known **security flaws**. This is a common and easily exploitable weakness if not managed.
- **Security Misconfigurations:** This is a broad category, including **default settings** that aren't secured, **unnecessary features** that are enabled, or **improper permissions** that expose sensitive information.
- **Cryptographic Failures:** Occurs when **sensitive data** (e.g., passwords, financial data) is not encrypted, is encrypted **poorly**, or is stored with weak hashing algorithms, leading to a high risk of **data exposure**.

Requirement of Security in Applications

- **Data Protection Imperatives**: Modern web applications handle sensitive personal information, financial data, and business-critical operations. Security breaches result in average costs of $4.45 million per incident, making robust protection essential for organizational survival.
- **Regulatory Compliance**: Applications must meet stringent requirements including GDPR, HIPAA, PCI-DSS, and SOX regulations. Non-compliance results in significant financial penalties and legal consequences.
- **Trust and Reputation**: Security breaches damage brand reputation and customer trust, with studies showing 60% of customers stop using services after data breaches.
- **Business Continuity**: Cyberattacks can disrupt operations for weeks, causing revenue loss and competitive disadvantage. Proactive security measures ensure operational resilience.
- **Evolving Threat Landscape**: Cybercriminals continuously develop sophisticated attack methods. AI-powered defenses are necessary to keep pace with automated attacks and advanced persistent threats.

Conclusion

In summary, this project successfully demonstrates the fusion of artificial intelligence and cybersecurity, culminating in a next-generation, self-learning security chatbot. By translating real-time network traffic and threat intelligence into actionable insights, our system moves beyond passive monitoring to:

- **Automate** proactive threat detection.
- **Enhance** overall network resilience.
- **Empower** users with immediate, AI-driven response actions.

This work establishes a robust and scalable blueprint for an enterprise-grade cybersecurity assistant, deliver intelligent, accessible security to businesses, educational institutions, and home networks alike.

References

1. AI in Cybersecurity: Latest Developments + How It's Used in 2025 - https://secureframe.com/blog/ai-in-cybersecurity
2. AI Dashboard Future-proofing Business Analytics: Design, Applications, and Latest Trends - https://fuselabcreative.com/ai-business-analytics-dashboard/
3. Vulnerabilities of Web Applications: Good Practices and New Trends - https://www.acigjournal.com/Vulnerabilities-of-Web-Applications-Good-Practices-and-New-Trends,199521,0,2.html
4. Web Application Scanning & API Security - https://www.qualys.com/apps/web-app-scanning/
5. Study of Vulnerability Scanners for Detecting SQL Injection and XSS Attack in Websites - https://ojs.bonviewpress.com/index.php/AIA/article/view/754/489
6. AI in Cybersecurity: Key Benefits, Defense Strategies, & Future Trends - https://www.fortinet.com/resources/cyberglossary/artificial-intelligence-in-cybersecurity