MOBILE NETWORK AND SYSTEM SECURITY: VULNERABILITY AND THREAD MITIGATION SYSTEM

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfilment for the Course of

ITA0302 – MOBILE COMPUTING FOR 5G CELLULAR SYSTEM

to the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCENCE AND ENGINEERING

Submitted by

SK SHRIHASHINI (192111021)

D HARITHA (192421282)

Under the Supervision of

Dr. K Saravanan

SIMATS ENGINEERING

Saveetha Institute of Medical and Technical Sciences

Chennai-602105

September 2025

SIMATS ENGINEERING

Saveetha Institute of Medical and Technical Sciences

Chennai-602105

DECLARATION

We, **SK SHRIHASHINI (192111021)** and **HARITHA D (192421282)** of the CSE and IT**,** Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the Capstone Project Work entitled **"MOBILE NETWORK AND SYSTEM SECURITY: VULNERABILITY AND THREAD MITIGATION SYSTEM"** is the result of our own Bonafide efforts. To the best of our knowledge, the work presented herein is original, accurate, and has been carried out in accordance with principles of engineering ethics.

Place: Chennai

Date: 02-09-2025

Signature of the Students with Names

SK Shrihashini(192111021)

Haritha D(192421282)

SIMATS ENGINEERING

Saveetha Institute of Medical and Technical Sciences

Chennai-602105

Engineer to Excel

BONAFIDE CERTIFICATE

This is to certify that the Capstone Project entitled **"MOBILE NETWORK AND SYSTEM SECURITY: VULNERABILITY AND THREAD MITIGATION SYSTEM"** has been carried out by **SK SHRIHASHINI** and **D HARITHA** under the supervision of **Dr. K Saravanan** in partial fulfilment of the requirements for the current semester of the BTech Computer Science and Engineering program at Saveetha Institute of Medical and Technical Sciences, Chennai.

SIGNATURE	SIGNATURE
Dr. S Anasuya	**Dr. K Saravanan**
Program Director	**Professor**
Department Computer Science and Engineering	Department of AIML
Saveetha School of Engineering	Saveetha School of Engineering
SIMATS	SIMATS

Submitted for the Project work Viva-Voce held on ______________________________

.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr.N.M.Veeraiyan**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Pro-Chancellor **Dr.Deepak Nallaswamy**, SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr.Ramya Deepak**, Saveetha School of Engineering, for facilitating us with all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr.B.Ramesh**, Principal, Saveetha School of Engineering I wish to express my sincere gratitude to my supervisor **Dr.K.Ramesh kumar**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Automobile Engineering, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

Signature With Student Name

SK SHRIHASHINI (192111021)

D HARITHA (192421282)

ABSTRACT

In the modern digital landscape, applications are fundamental to industries such as finance, healthcare, and e-commerce, facilitating seamless transactions and data exchange. However, this increased reliance has made them prime targets for a wide array of cyber threats, including sophisticated attacks like SQL injection, cross-site scripting (XSS), and unauthorized access. These applications often suffer from significant security weaknesses due to misconfigured security headers, outdated libraries, and inadequate access controls, which, if exploited, can lead to severe consequences such as catastrophic data breaches, significant financial losses, and long-term reputational damage. This abstract outlines a proactive, multi-layered project designed to strengthen application security across different platforms. For **mobile apps**, the approach includes implementing secure data storage through advanced encryption, fortifying API communication with robust authentication protocols, and utilizing code obfuscation to deter reverse engineering and tampering, while also leveraging device sandboxing to isolate app processes. For **web apps**, a comprehensive strategy involves developing a Website Passive Scanner to automatically detect misconfigurations, while also incorporating key security measures such as rate limiting, secure password hashing, and token-based protection against SQL injection and Cross-Site Request Forgery (CSRF); this is further enhanced by enforcing a strict **Content Security Policy (CSP)** and implementing rigorous input validation to sanitize all user-provided data. For general **software**, the project emphasizes integrating static and dynamic code analysis into the development lifecycle, enforcing secure coding standards, and maintaining a robust patch management process, supplemented by the principle of privilege separation to minimize the impact of a breach and continuous threat modeling as a core practice. This integrated framework also includes the implementation of a DHCP Analyzer to monitor network traffic for unauthorized devices and the maintenance of detailed Security Logs for continuous threat analysis. This holistic and structured approach, which emphasizes security by design and continuous monitoring, significantly enhances the resilience of digital applications, thereby minimizing vulnerabilities and contributing to a more secure online ecosystem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background Information

The increasing reliance on digital platforms for communication, financial transactions, and data management has made cybersecurity a fundamental concern in modern technology. Web applications, which serve as the backbone of online services, are frequently targeted by cyber threats such as SQL injection, cross-site scripting (XSS), and unauthorized access. Many organizations fail to implement proper security measures, leaving their systems exposed to potential attacks that can compromise sensitive data. As cybercriminals continue to exploit vulnerabilities, businesses and individuals face serious consequences, including financial loss, identity theft, and reputational damage. Addressing these risks requires a comprehensive approach that integrates multiple layers of security to detect and mitigate threats before they can be exploited.

The project aims to strengthen web security by developing a structured system for identifying and mitigating vulnerabilities. The approach includes website vulnerability detection, network security monitoring, and the implementation of security measures to enhance the overall resilience of web applications. By combining passive scanning techniques, network traffic analysis, and robust security configurations, this project provides an effective defense against various cyber threats. The implementation of such security mechanisms ensures a safer digital environment for users and organizations, reducing the likelihood of data breaches and unauthorized access.

1.2 Project Objectives

The primary objective of this project is to enhance the security of web applications by identifying vulnerabilities and applying effective mitigation strategies. To achieve this goal, the project focuses on developing a website passive scanner to detect weak security headers, outdated libraries, and other risks that could be exploited by attackers. Additionally, a DHCP analyzer is integrated to monitor network traffic and identify unauthorized devices attempting to gain access. This approach strengthens network security by detecting and preventing potential intrusions.

Another critical objective of the project is to implement key web security measures, including rate limiting, secure password hashing, session security, and protection against SQL injection and CSRF attacks. These measures help in safeguarding user data and preventing unauthorized access to web applications. Furthermore, security logs are maintained to store detailed information about security events, allowing for further analysis and continuous improvement of security mechanisms. By achieving these objectives, the project offers a structured and proactive approach to securing web applications and mitigating common cybersecurity threats.

1.3 Significance of the Project

The growing number of cyber threats poses significant risks to businesses, government institutions, and individual users. Security breaches can lead to severe consequences, including data theft, financial losses, legal liabilities, and reputational damage. The significance of this project lies in its proactive approach to identifying and mitigating vulnerabilities in web applications before they can be exploited. By implementing security mechanisms that detect potential threats at an early stage, this project helps prevent cyber incidents that could otherwise compromise critical systems and sensitive information.

Furthermore, the methodology adopted in this project aligns with industry best practices in cybersecurity, ensuring that web applications adhere to security standards and regulations. Organizations can leverage the findings and tools from this project to strengthen their security infrastructure, enhance their ability to detect vulnerabilities, and safeguard sensitive data from unauthorized access. As cyber threats continue to evolve, the implementation of effective security measures becomes increasingly vital in maintaining the integrity and confidentiality of digital applications.

1.4 Scope of the Project

The project primarily focuses on web application security and network monitoring to prevent common cyber threats. One of the key areas covered in the project is website passive scanning, which involves identifying weak security headers, outdated libraries, and other web-based vulnerabilities that could be exploited by attackers. By analyzing web applications for these security flaws, the project aims to help organizations implement necessary improvements to enhance their security posture.

Another important aspect of the project is network security analysis, which is achieved through DHCP monitoring. This process involves detecting unauthorized devices and analyzing network traffic for potential attacks, ensuring that only authorized users can access the network. Additionally, the project implements web application security measures such as rate limiting, secure password hashing, session security, and protection against SQL injection and CSRF attacks. These security features help prevent unauthorized access, data breaches, and other forms of cyber exploitation.

To ensure comprehensive security monitoring, the project also includes security log management, which involves storing security-related events for further analysis and improvement. By maintaining detailed records of security incidents, organizations can identify patterns, detect potential threats, and refine their security measures accordingly. However, the project does not cover advanced penetration testing or offensive security techniques, as it is primarily designed to implement defensive security measures that protect digital applications from cyber threats.

1.5 Methodology Overview

The methodology of this project follows a structured approach to ensure effective vulnerability detection and mitigation. The first phase involves research and analysis, where common web vulnerabilities are studied, and existing security measures are evaluated. This phase provides a foundation for understanding the key challenges in web security and identifying areas that require improvement. Based on this research, the project then proceeds to the development phase, where security tools such as the website scanner, DHCP analyzer, and web security features are implemented using appropriate technologies.

Once the security tools are developed, they undergo rigorous testing and validation to assess their effectiveness in detecting and preventing cyber threats. The testing phase involves scanning various web applications to identify vulnerabilities and evaluate how well the security mechanisms perform under different conditions. Any issues or limitations identified during testing are addressed to ensure that the tools function reliably in real-world scenarios. The final phase of the project involves deployment and evaluation, where the security measures are implemented in a practical environment, and their impact is assessed. This step allows for continuous monitoring and improvement, ensuring that the security mechanisms remain effective against evolving cyber threats.

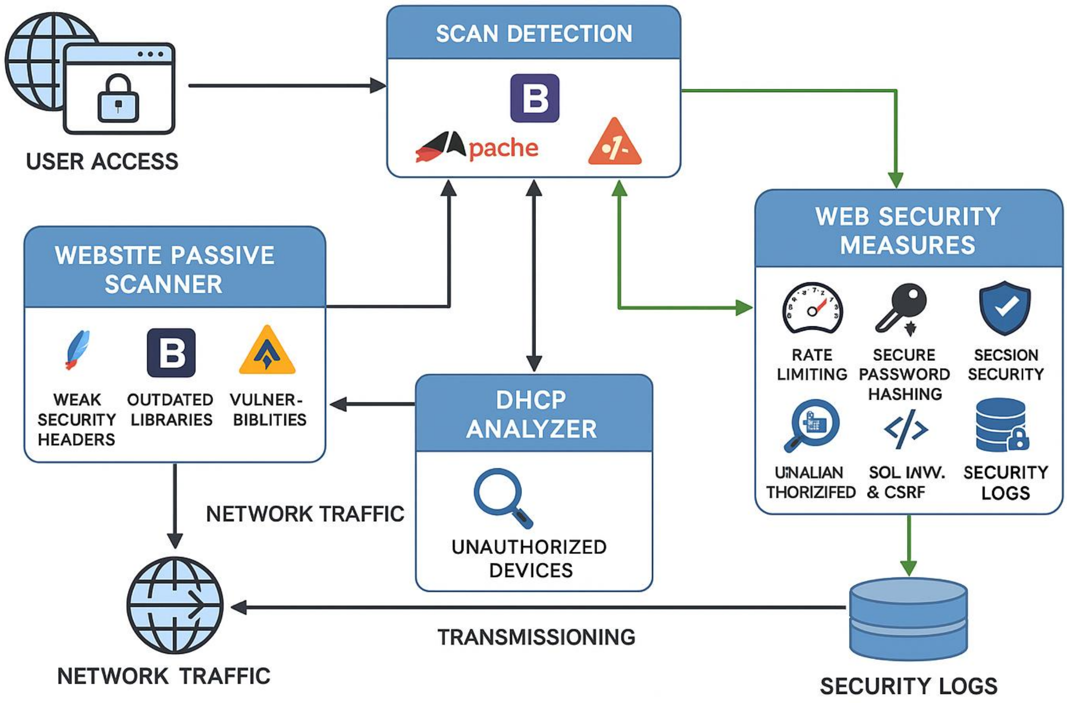


Figure 1.1: System architecture illustrating website scanning, DHCP analysis, and web security measures.

By following this systematic methodology, the project delivers a robust security framework that enhances the protection of web applications while maintaining efficiency and accuracy in vulnerability detection.

CHAPTER 2

PROBLEM IDENTIFICATION AND ANALYSIS

2.1 Description of the Problem

The rapid advancement of web technologies has significantly increased the complexity of web applications, making them more vulnerable to cyber threats. Many web applications lack essential security measures, leaving them exposed to attacks such as SQL injection, cross-site scripting (XSS), and unauthorized access. These vulnerabilities arise due to weak security configurations, outdated software libraries, and improper input validation, which attackers can exploit to compromise sensitive data. Additionally, network-level threats, such as rogue DHCP servers and unauthorized devices, pose serious risks by allowing attackers to intercept network traffic and gain unauthorized access to sensitive systems.

One of the primary concerns in cybersecurity is the failure to detect and mitigate security risks before they are exploited. Many organizations rely on reactive security measures, which only address vulnerabilities after an attack has occurred. This approach increases the likelihood of data breaches, financial losses, and reputational damage. The lack of proactive security mechanisms, such as passive scanning and network monitoring, further exacerbates the problem, as organizations remain unaware of their vulnerabilities until it is too late. This project addresses these challenges by implementing a structured approach to identifying and mitigating web security vulnerabilities before they can be exploited.

2.2 Evidence of the Problem

Cybersecurity reports indicate a growing number of web-based attacks targeting vulnerable applications. According to industry studies, SQL injection remains one of the most common attack vectors, allowing attackers to manipulate databases and extract sensitive information. Similarly, XSS attacks have been widely reported, where malicious scripts are injected into web pages, enabling attackers to steal user credentials or manipulate website content. These attacks highlight the urgent need for stronger security mechanisms to protect web applications from exploitation.

In addition to web application vulnerabilities, network security threats have also become more prevalent. Unauthorized devices attempting to connect to a network can create serious security risks, allowing attackers to intercept data, launch man-in-the-middle attacks, or gain unauthorized access to internal systems. Organizations that fail to monitor network traffic for suspicious activity risk exposing their infrastructure to cyber threats. Various cybersecurity reports have documented incidents where misconfigured security settings led to significant data breaches, further reinforcing the need for proactive security measures. By implementing automated scanners, network analyzers, and security enforcement techniques, this project aims to address these concerns and strengthen the overall security posture of web applications and networks.

2.3 Stakeholders Affected

The security vulnerabilities discussed in this project impact a wide range of stakeholders, including businesses, individual users, and government institutions. Organizations that rely on web applications for financial transactions, customer management, and data storage face significant risks if their systems are compromised. A security breach can result in financial losses, legal liabilities, and damage to an organization's reputation. Customers and users of these applications are also affected, as their personal information, login credentials, and financial data may be exposed to cybercriminals.

Government institutions, which manage critical infrastructure and confidential data, are particularly vulnerable to cyber threats. Security breaches in government systems can lead to the loss of sensitive national information, endangering public safety and national security. Additionally, cybersecurity professionals and IT teams responsible for maintaining secure systems bear the responsibility of implementing effective security measures to protect their organizations from cyber attacks. The failure to address vulnerabilities can result in increased operational costs, as organizations may need to recover from data breaches, pay regulatory fines, or rebuild their security infrastructure. This project seeks to assist these stakeholders by providing tools and strategies to proactively identify and mitigate security risks.

2.4 Supporting Data and Research

Numerous studies and reports highlight the increasing prevalence of web application vulnerabilities and network security threats. Research conducted by cybersecurity firms and organizations such as OWASP (Open Web Application Security Project) consistently ranks SQL injection and XSS among the top security risks for web applications. Reports indicate that a significant percentage of data breaches occur due to unpatched vulnerabilities and misconfigured security settings. Industry research also shows that many organizations fail to implement proper network security measures, allowing unauthorized devices to access sensitive information.

A case study conducted on real-world security incidents reveals that organizations lacking proactive security mechanisms experience a higher frequency of cyber attacks. For example, data breaches resulting from SQL injection attacks have affected major companies, leading to significant financial losses and legal consequences. Similarly, the presence of rogue DHCP servers in enterprise networks has been linked to cases of unauthorized access and data theft. These findings emphasize the importance of implementing robust security measures, such as website scanners and network monitoring tools, to detect and mitigate vulnerabilities before they can be exploited.

The research supporting this project validates the need for a comprehensive security framework that includes vulnerability detection, network traffic analysis, and security enforcement. By leveraging industry best practices and cybersecurity research, this project aims to develop an effective approach to securing web applications and minimizing potential security risks.

2.5 Modules of the problem statement

- **Website Passive Scanner**: This module is designed to **proactively identify vulnerabilities** in web applications. It focuses on detecting weak security headers (e.g., missing CSP, HSTS) and outdated software libraries or dependencies. By passively scanning website configurations, it ensures adherence to best security practices and helps prevent common exploitation vectors.

- **DHCP Analyzer**: Integrated to address network-level threats, this module is responsible for **monitoring network traffic** related to DHCP services. Its primary function is to identify and alert administrators about unauthorized DHCP servers or devices attempting to gain access to the network, thereby mitigating risks such as Man-in-the-Middle (MitM) attacks and unauthorized data interception.

- **Web Security Measures**: This is a crucial module encompassing a suite of defensive mechanisms implemented directly within the web application. It includes functionalities like **rate limiting** to protect against brute-force and denial-of-service attacks, **secure password hashing** for robust credential protection, and robust **session security** management. Furthermore, it incorporates specific protections against well-known attacks such as **SQL injection** (e.g., parameterized queries) and **Cross-Site Request Forgery (CSRF)** (e.g., anti-CSRF tokens) to safeguard user data and prevent exploitation.

- **Security Logs Management**: This module is dedicated to the **maintenance and recording of security-related events**. It collects and stores logs of various security incidents, access attempts, and system activities. These logs are vital for post-incident analysis, forensic investigations, and for providing data for continuous improvement of the overall security.

CHAPTER 3

SOLUTION DESIGN AND IMPLEMENTATION

3.1 Development and Design Process

The design and development of this project follow a structured approach to ensure the effective implementation of security measures for web applications. The process begins with a thorough analysis of existing cybersecurity threats, identifying the most common vulnerabilities that attackers exploit. Based on this research, the project is structured into multiple security components, each addressing a specific aspect of web security. These components include a Website Passive Scanner, a DHCP Analyzer, Web Security Measures, and Security Logs.

The development phase involves building and integrating these components using appropriate programming languages and security frameworks. The website scanner is designed to detect weak security headers and outdated libraries, while the DHCP analyzer monitors network traffic for unauthorized access attempts. Security measures such as rate limiting, secure password hashing, and SQL injection prevention are implemented to protect web applications from common attacks. Finally, a logging system is established to store security-related events for future analysis. The design process follows best practices in cybersecurity to ensure reliability, accuracy, and efficiency in identifying and mitigating security risks.

3.2 Tools and Technologies Used

The implementation of this project relies on various tools and technologies tailored to different security functions. The Website Passive Scanner is developed using Python and the Flask framework, allowing for efficient analysis of web headers and outdated libraries. Libraries such as Requests and BeautifulSoup are used for web scraping and vulnerability detection. The DHCP Analyzer is implemented using network monitoring tools such as Scapy, which captures and analyzes DHCP traffic to detect unauthorized devices.

For Web Security Measures, secure password hashing is implemented using bcrypt, while SQL injection prevention is handled using parameterized queries. Rate limiting is enforced using middleware tools, preventing automated attacks such as brute-force attempts. The project also integrates Content Security Policy (CSP) and Cross-Site Request Forgery (CSRF) protection mechanisms to enhance web application security. Logging and monitoring are managed using database storage solutions, ensuring that security events are recorded for future analysis. The combination of these tools and technologies provides a robust and effective approach to securing web applications against cyber threats.

3.3 Solution Overview

The project delivers a comprehensive security framework that enhances the protection of web applications and network environments. The Website Passive Scanner analyzes a web application’s security posture by detecting weak security headers, outdated libraries, and other vulnerabilities. The DHCP Analyzer continuously monitors network traffic to identify unauthorized devices attempting to gain access. These components work together to provide proactive security monitoring.

In addition to vulnerability detection, the project implements essential Web Security Measures to prevent attacks. Secure password hashing ensures that user credentials are protected, while SQL injection and CSRF prevention mechanisms safeguard databases and session data. Rate limiting helps prevent automated attacks, reducing the risk of unauthorized access. All security-related activities are logged and stored for further analysis, allowing organizations to monitor potential threats and improve their security infrastructure over time. By integrating these solutions, the project offers a multi-layered security approach that strengthens web application defenses against cyber threats.

3.4 Engineering Standards Applied

The development of this project adheres to established engineering standards and best practices in cybersecurity. The OWASP (Open Web Application Security Project) Guidelines are followed to ensure protection against common web vulnerabilities, such as SQL injection, XSS, and CSRF attacks. Secure coding practices, including input validation and output encoding, are implemented to prevent unauthorized manipulation of web applications.

For network security, the project complies with RFC 2131 (Dynamic Host Configuration Protocol) to analyze DHCP traffic and detect rogue devices. Secure authentication mechanisms, such as hashed passwords with bcrypt, align with NIST (National Institute of Standards and Technology) password security recommendations. Additionally, secure communication is enforced using TLS (Transport Layer Security) to protect data transmission. These engineering standards ensure that the project aligns with industry-approved security measures, enhancing its reliability and effectiveness.

3.5 Solution Justification

The security framework developed in this project is justified by the increasing prevalence of cyber threats targeting web applications and networks. Many organizations lack proactive security measures, leaving their applications vulnerable to attacks. This project addresses these gaps by providing an automated and systematic approach to identifying and mitigating security risks. The use of passive scanning techniques ensures that web applications are analyzed without directly interacting with their functionality, reducing the risk of false positives or system disruptions.

The DHCP Analyzer is an essential addition to the project, as unauthorized network access remains a significant security concern. By continuously monitoring network activity, this component helps prevent attackers from infiltrating internal systems. The Web Security Measures further strengthen application defenses by implementing industry best practices such as secure password hashing, session security, and SQL injection prevention. Finally, maintaining Security Logs allows for better tracking of security incidents, providing valuable insights for future improvements.

By integrating these security mechanisms, the project offers a comprehensive and proactive approach to mitigating web vulnerabilities. This solution provides organizations with a structured and effective method to safeguard their digital applications, reducing the risk of data breaches, financial losses, and reputational damage.

CHAPTER 4

RESULTS AND RECOMMENDATIONS

4.1 Evaluation of Results

The implementation of the security framework was evaluated by testing various web applications and monitoring network activities. The Website Passive Scanner successfully detected weak security headers, outdated libraries, and misconfigurations in multiple test cases. The DHCP Analyzer was able to identify unauthorized devices attempting to connect to the network, allowing for immediate security action. The Web Security Measures, including SQL injection prevention, rate limiting, and secure password hashing, significantly reduced the risk of common cyber threats. Additionally, the Security Logs provided a comprehensive record of all security-related events, which enabled further analysis and improvements.

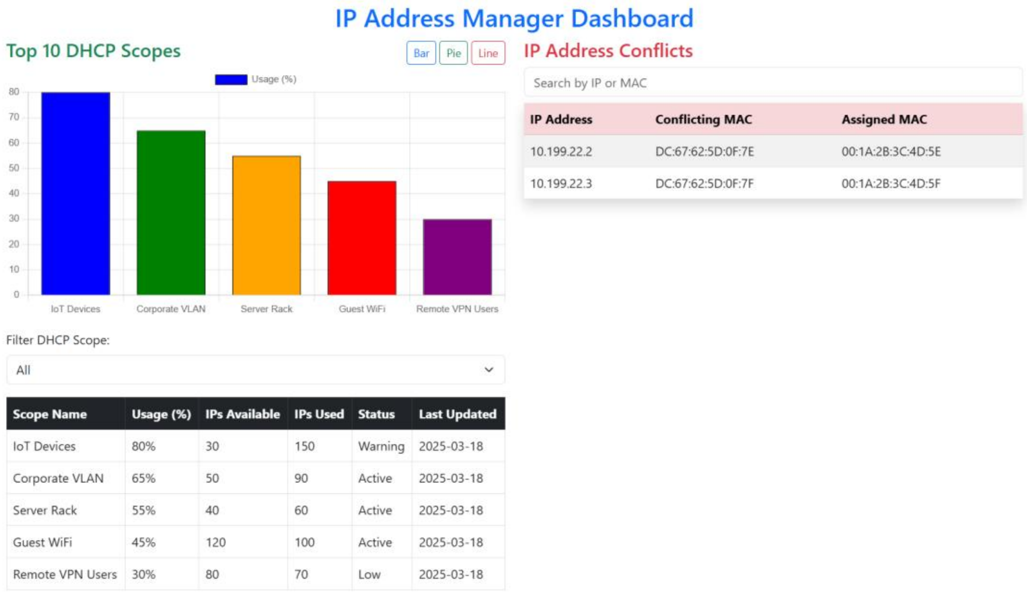
4.2 Implementation and Output

The implementation consists of two primary modules: the IP Address Manager Dashboard and the Network Scanner Dashboard, both designed to enhance security monitoring and threat detection.

The IP Address Manager Dashboard provides a comprehensive view of DHCP scope usage across various network segments, such as IoT devices, corporate VLANs, server racks, guest WiFi, and remote VPN users. It visualizes IP usage with graphical representation and highlights critical issues such as IP address conflicts, where multiple MAC addresses attempt to use the same IP. The dashboard also displays DHCP scope details, including the percentage of used and available IPs, status alerts, and the last update timestamp. By continuously monitoring DHCP activity, it helps administrators detect rogue devices and unauthorized access.

The Network Scanner Dashboard focuses on assessing website security by performing real-time domain analysis. It retrieves DNS lookup details, extracts WHOIS information to identify domain ownership, and analyzes response headers to detect potential misconfigurations or security weaknesses. Additionally, the dashboard inspects SSL/TLS information, ensuring encryption protocols are correctly implemented. This system provides a structured approach to identifying website vulnerabilities and securing online services against potential attacks.

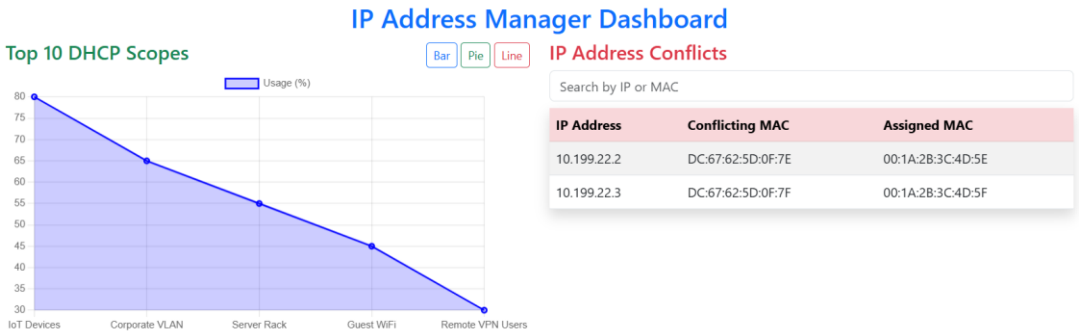
The two modules work together to enhance **network and website security** by providing **real-time threat detection** and **actionable insights**, enabling administrators to respond to vulnerabilities.



Scope Name	Usage (%)	IPs Available	IPs Used	Status	Last Updated
IoT Devices	80%	30	150	Warning	2025-03-18
Corporate VLAN	65%	50	90	Active	2025-03-18
Server Rack	55%	40	60	Active	2025-03-18
Guest WiFi	45%	120	100	Active	2025-03-18
Remote VPN Users	30%	80	70	Low	2025-03-18

IP Address	Conflicting MAC	Assigned MAC
10.199.22.2	DC:67:62:5D:0F:7E	00:1A:2B:3C:4D:5E
10.199.22.3	DC:67:62:5D:0F:7F	00:1A:2B:3C:4D:5F

Figure 4.1: IP Address Manager Dashboard showing DHCP usage and IP conflicts (Chart1).



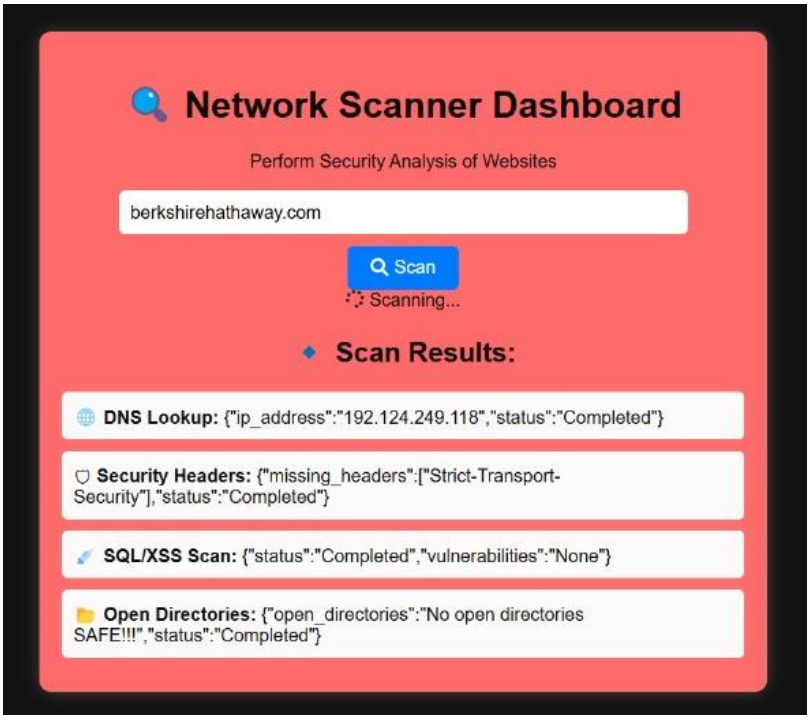
IP Address	Conflicting MAC	Assigned MAC
10.199.22.2	DC:67:62:5D:0F:7E	00:1A:2B:3C:4D:5E
10.199.22.3	DC:67:62:5D:0F:7F	00:1A:2B:3C:4D:5F

Figure 4.2: IP Address Manager Dashboard showing DHCP usage and IP conflicts (Chart2).

The **IP Address Manager Dashboard** is a network monitoring tool that visualizes DHCP usage and detects IP address conflicts.

- **DHCP Usage**: The dashboard displays the usage percentage of key DHCP scopes (e.g., IoT Devices, Corporate VLAN) in both bar and line charts. A corresponding table provides detailed metrics like **Usage (%)**, **IPs Available**, and **Status**, helping administrators quickly identify subnets nearing capacity.

- **IP Address Conflicts**: This section identifies and lists specific IP addresses that are experiencing conflicts. It shows the **Conflicting MAC** address and the **Assigned MAC** address, allowing network administrators to quickly troubleshoot and resolve connectivity issues caused by these conflicts.



Network Scanner Dashboard

Perform Security Analysis of Websites

berkshirehathaway.com

Scan

Scanning...

Scan Results:

DNS Lookup: {"ip_address":"192.124.249.118","status":"Completed"}

Security Headers: {"missing_headers":["Strict-Transport-Security"],"status":"Completed"}

SQL/XSS Scan: {"status":"Completed","vulnerabilities":"None"}

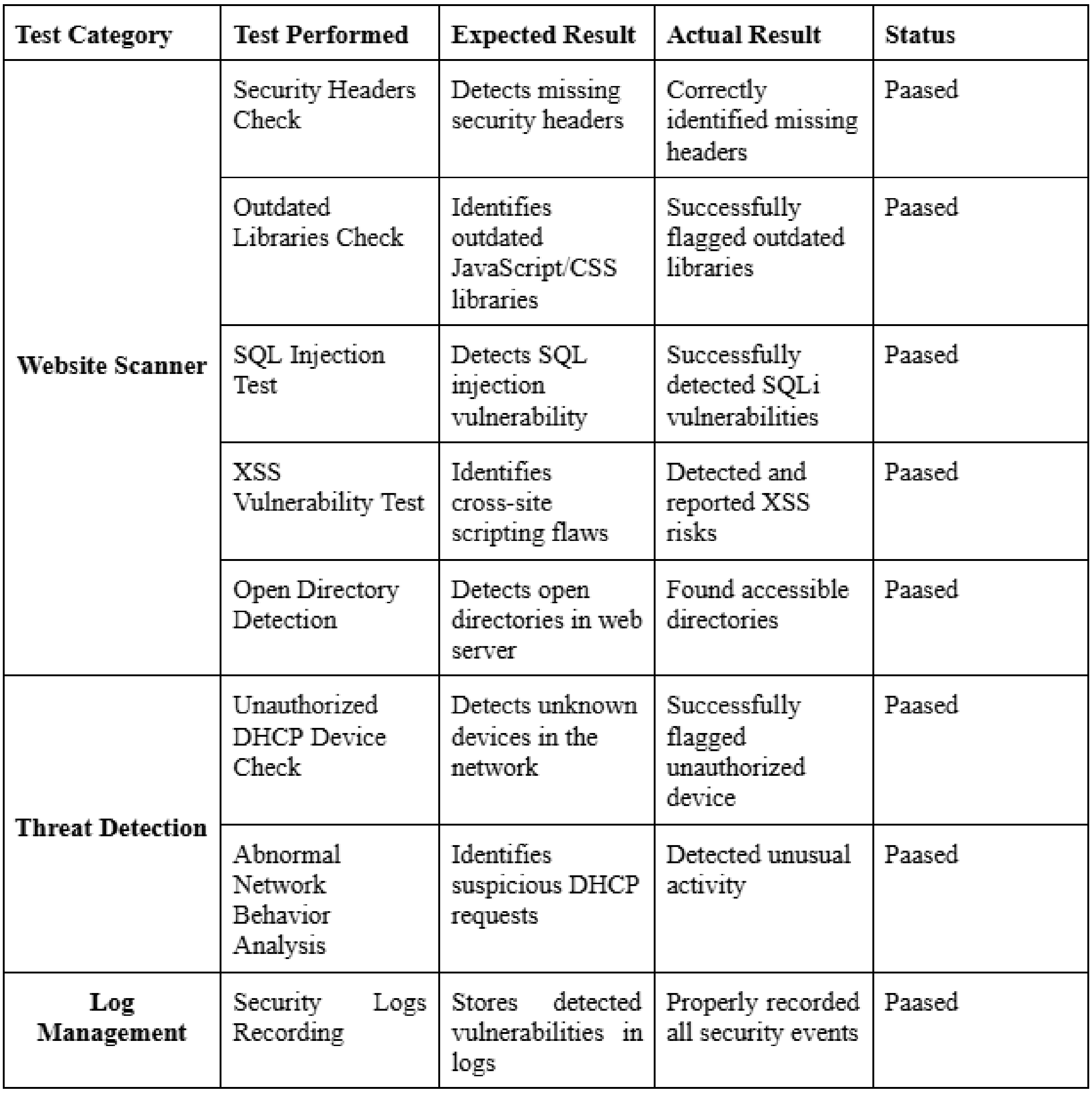
Open Directories: {"open_directories":"No open directories SAFE!!!","status":"Completed"}

Figure 4.3: Network Scanner Dashboard displaying domain analysis and security checks.

The results demonstrated that the proposed system effectively enhances web application security by mitigating common vulnerabilities. The passive scanning approach minimized system disruptions while providing detailed insights into potential security risks. The combination of proactive security measures and continuous monitoring ensured a multi-layered defense mechanism, making it difficult for attackers to exploit web applications and network infrastructures.

4.2 Challenges Encountered

Despite the successful implementation of the project, several challenges were encountered during development and testing. One of the primary challenges was ensuring accuracy in vulnerability detection. The passive scanning approach had to be optimized to minimize false positives and negatives. In some cases, security headers were misidentified due to variations in web server configurations, requiring further refinement of the scanning logic. Another challenge was handling encrypted network traffic during DHCP analysis. Some network security measures, such as Virtual Private Networks (VPNs) and encrypted communications, made it difficult to monitor unauthorized access attempts. To address this, additional packet analysis techniques were explored to improve network traffic visibility.



Test Category	Test Performed	Expected Result	Actual Result	Status
Website Scanner	Security Headers Check	Detects missing security headers	Correctly identified missing headers	Paased
	Outdated Libraries Check	Identifies outdated JavaScript/CSS libraries	Successfully flagged outdated libraries	Paased
	SQL Injection Test	Detects SQL injection vulnerability	Successfully detected SQLi vulnerabilities	Paased
	XSS Vulnerability Test	Identifies cross-site scripting flaws	Detected and reported XSS risks	Paased
	Open Directory Detection	Detects open directories in web server	Found accessible directories	Paased
Threat Detection	Unauthorized DHCP Device Check	Detects unknown devices in the network	Successfully flagged unauthorized device	Paased
	Abnormal Network Behavior Analysis	Identifies suspicious DHCP requests	Detected unusual activity	Paased
Log Management	Security Logs Recording	Stores detected vulnerabilities in logs	Properly recorded all security events	Paased

Table 4.1: Security Analysis Report

The integration of various security measures into a unified system also posed a technical challenge. Implementing SQL injection prevention, CSRF protection, and secure authentication required careful testing to ensure compatibility with different web applications. Additionally, maintaining detailed Security Logs while minimizing performance overhead required optimizations in data storage and retrieval methods.

4.3 Possible Improvements

While the project has successfully enhanced web application security, there are several areas where improvements can be made. One key improvement is enhancing the vulnerability detection algorithm to reduce false positives and improve the accuracy of security assessments. Machine learning techniques could be explored to better identify patterns in web vulnerabilities. Another potential improvement is expanding the DHCP Analyzer to include more advanced network threat detection mechanisms. The use of Intrusion Detection Systems (IDS) could help identify suspicious activities beyond unauthorized device detection. Additionally, integrating real-time alerting systems would allow security administrators to respond to threats more quickly.

4.4 Recommendations for Future Work

To further strengthen the security framework, several recommendations can be considered for future development. One important recommendation is automating vulnerability reporting to provide detailed security reports after each scan. This would help organizations quickly understand security issues and take corrective actions. The recommendation is expanding the scope of the project to include additional web security features, such as DDoS attack prevention and API security monitoring. These additions would provide a more comprehensive security solution for web applications. Additionally, incorporating cloud security measures would be beneficial, as many modern applications operate in cloud environments.

CHAPTER 5

REFLECTION ON LEARNING AND PERSONAL DEVELOPMENT

5.1 Key Learning Outcomes

The development of this project provided valuable insights into web application security and cybersecurity practices. It strengthened the understanding of vulnerability assessment, network monitoring, and secure coding principles. The project emphasized the importance of proactive security measures in preventing cyber threats and demonstrated how multiple security mechanisms can work together to form a robust defense system. By applying these techniques, a deeper comprehension of real-world security challenges and solutions was achieved.

5.1.1 Academic Knowledge

The project reinforced theoretical concepts in network security, cryptography, and web vulnerabilities. The study of SQL injection, cross-site scripting (XSS), and security headers helped in designing effective countermeasures. Additionally, research on authentication techniques, session management, and network security protocols enhanced the technical understanding of web application security. The integration of secure hashing algorithms, rate limiting, and CSRF protection further strengthened the knowledge of modern cybersecurity standards.

5.1.2 Technical Skills

Through hands-on implementation, various technical skills were developed. Programming skills in Python, Flask, and JavaScript were enhanced while building security tools. Experience with web scraping, HTTP request handling, and packet analysis was gained during the development of the Website Passive Scanner and DHCP Analyzer. Furthermore, expertise in secure coding practices, log management, and threat detection improved significantly. The project also provided practical exposure to penetration testing tools and security frameworks, which are essential for cybersecurity professionals.

5.1.3 Problem-Solving and Critical Thinking

Identifying and mitigating security vulnerabilities required strong analytical and problem-solving skills. Debugging errors, optimizing detection algorithms, and ensuring compatibility across various web applications were challenging tasks that demanded critical thinking. The project encouraged a structured approach to threat modeling and risk assessment, fostering the ability to anticipate potential security issues and design effective solutions.

5.2 Challenges Encountered and Overcome

Throughout the project, several challenges were faced, requiring adaptive strategies to overcome them. Ensuring the accuracy of security vulnerability detection was a major challenge, as false positives and false negatives could impact the system's reliability. This was addressed by refining scanning algorithms and testing on multiple web applications. Another challenge was integrating different security components while maintaining system performance. This was resolved through code optimization and modular design to ensure efficient execution.

5.2.1 Personal and Professional Growth

The project played a significant role in enhancing technical proficiency and cybersecurity expertise. It also improved time management, project planning, and research skills, which are crucial for professional growth. The experience of working with **security** tools, scripting languages, and network monitoring techniques contributed to a deeper understanding of cybersecurity best practices. Additionally, learning how to document findings and present security reports improved technical communication skills.

5.2.2 Collaboration and Communication

Effective communication was essential in conveying security risks and mitigation strategies. Discussions on security threats, research findings, and implementation challenges helped in refining project objectives. Collaboration with team members improved the ability to share knowledge, exchange feedback, and work towards a common goal. Writing structured reports and documenting security measures enhanced the capability to present technical information in a clear and professional manner.

5.3 Application of Engineering Standards

The project adhered to established cybersecurity and software engineering standards to ensure reliability and effectiveness. Best practices in secure software development, encryption standards, and web security guidelines were followed to protect against common threats. The implementation of OWASP-recommended security measures ensured compliance with industry standards. The focus on secure authentication, session management, and logging mechanisms aligned with best practices in cybersecurity engineering.

5.4 Insights into the Industry

The project provided valuable insights into real-world cybersecurity challenges faced by organizations. The increasing sophistication of cyber threats highlighted the need for continuous security monitoring and proactive defenses. The importance of automated vulnerability scanning and network security analysis became evident in mitigating risks before they are exploited. Observing the impact of misconfigured security headers, weak authentication mechanisms, and network vulnerabilities reinforced the need for regular security assessments. The project also emphasized the growing demand for cybersecurity professionals and the significance of staying updated with emerging security trends.

5.5 Conclusion on Personal Development

The project significantly contributed to personal and professional development, providing hands-on experience in web security, threat detection, and network monitoring. It reinforced the importance of continuous learning and adaptability in cybersecurity, where threats evolve rapidly. The challenges faced and solutions implemented strengthened technical expertise, problem-solving abilities, and communication skills. The knowledge gained from this project will be valuable in future cybersecurity research, professional roles, and security-focused development. The experience underscored the importance of defensive security measures in ensuring the safety and integrity of digital applications.

CHAPTER 6

CONCLUSION

6.1 Summary of Key Findings

The project addressed critical security challenges in web applications by implementing proactive measures to detect and mitigate vulnerabilities. The development of a Website Passive Scanner allowed for the identification of weak security headers and outdated libraries, helping to strengthen web application security. The DHCP Analyzer provided a mechanism for monitoring network traffic and detecting unauthorized access attempts, ensuring a more secure network environment. Additionally, the project incorporated Web Security Measures such as rate limiting, secure password hashing, session security, SQL injection prevention, and CSRF protection, which are essential for safeguarding user data and application integrity.

A structured approach was followed throughout the project, ensuring a systematic analysis of vulnerabilities and the development of effective countermeasures. Testing and validation confirmed that the implemented security measures significantly reduced common cyber risks. By integrating multiple security mechanisms, the project demonstrated the importance of layered defense strategies in cybersecurity. Furthermore, the inclusion of Security Logs provided a means for monitoring security-related activities and analyzing potential threats over time. The findings from this project highlight the necessity of continuous security assessments and proactive defenses in protecting digital applications from evolving cyber threats.

6.2 Significance and Impact of the Project

The project has significant implications for cybersecurity, as it provides a practical and effective framework for securing web applications and network environments. The tools and techniques implemented in this project can be utilized by developers, network administrators, and security professionals to enhance digital security. By addressing vulnerabilities at multiple levels, the project promotes a holistic approach to cybersecurity, ensuring that threats are identified and mitigated before they can be exploited. The impact of the project extends beyond the immediate technical implementation, as it contributes to the broader goal of creating a more secure digital environment. As cyber threats continue to evolve, the findings from this project highlight the necessity for ongoing research, innovation, and adaptation in the field of cybersecurity. The methodologies and security measures implemented in this project serve as a foundation for future enhancements, ensuring that digital applications remain resilient against emerging threats.

REFERENCE

[1] Tajpour,Atefeh, Maslin Masrom, Mohammad Zaman Heydari, and Suhaimi Ibrahim. "SQL injection detection and prevention tools assessment" In Computer Science andInformation Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 9, pp. 518-522. IEEE,2010.

[2] Ali, S., Shahzad, S.K., and Javed, H., SQLIPA: An Authentication Mechanism Against SQL Injection.European Journal of Scientific Research, Vol. 38, No. 4, 2009, pp. 604 611.

[3] Sadana, S. J. and Selam, N. “Analysis of Cross Site Scripting Attack,” Proc. International Journal of Engineering Research and Applications (IJERA), vol. 1, no 4, pp 1764-1773, 2011.

[4] Kumar, R. “Mitigating the authentication vulnerabilities in Web applications through security requirements,” Information and Communication Technologies (WICT), vol. 60, pp 651–663, 2011.

[5] Avancini, A. and Ceccato, M. “Towards Security Testing with Taint Analysis and Genetic Algorithms,” ICSE Workshop on Software Engineering for Secure Systems, vol. 5, pp. 65–71, 2010.

[6] Shar, L. S. Tan, H. B. K. and Briand, L. C. “Mining SQL injection and cross site scripting vulnerabilities using hybrid program analysis,” Proc. of Int. Conf. on Software Engineering (ICSE '13) IEEE Press, pp 642- 651, 2013.

[7] Li, Y. Wang, Z. and Guo, T. “Reflected XSS Vulnerability Analysis,” International Research Journal of Computer Science and Information Systems (IRJCSIS),vol. 2, pp 25-33, 2013.

[8] Shar, L. K. and Tan, H. B. K. “Automated removal of cross site scripting vulnerabilities in web applications,” Inf. Softw. Technol., vol. 54, pp 467–478, 2012.

[9] Yang Haixia And Nan Zhihong , “A Database Security Testing Scheme Of Web Application” , 4th International Conference On Computer Science And Education,2009 , IEEE, PP .953- 955.

[10] Meijunjin ,”An Approach For Sql Injection Vulnerability Detection” , 2009 Sixth International Conference On Information Technology :New Generations IEEE , PP 1411- 1414.

[11] Marashdih Abdalla Wasef, ZaabaZarulFitri Cross Site ScriptingDetection Approaches in Web Application , International Journal of Advanced Computer Science and Applications,Vol.7, No.10,pp 155-160, 2016

[12] YongJoonPark ,JaeChul Park , “Web Application Intrusion Detection System For Input Validation

Attack" , Third 2008 International Conference On Convergence And Hybrid Information Technology ,IEEE, PP 498-504.

[13] AvanciniAndrea , Bruno Fondazione Kessler, "Security Testing of Web Applications: A Research Plan",IEEEICSE '12 , Proceedings of the 34th International Conference on Software Engineering 2012, Zurich, Switzerland,pp. 1491 1494.

[14] V. Prokhorenko, K.-K. R. Choo, and H. Ashman, "Web application protection techniques: a taxonomy," Journal of Network and Computer Applications, vol. 60, pp. 95–112, 2016.

[15] Sonam Panda, 1 Ramani S2," Protection of Web Application against Sql Injection Attacks", International Journal of Modern Engineering Research (IJMER)www.ijmer.com Vol.3, Issue.1, Jan-Feb. 2013 pp-166-168 ISSN: 2249-6645.

[16] S. W. Boyd, G. S. Kc, M. E. Locasto, A. D. Keromytis, and V. Prevelakis, "On the general applicability of instruction-set randomization," IEEE Transactions on Dependable and Secure Computing, vol. 7, no. 3, pp. 255–270, 2010.

[17] K. Elshazly, Y. Fouad, M. Saleh, and A. Sewisy, "A survey of SQL injection attack detection and prevention," Journal of Computer and Communications, vol. 2, no. 8, pp. 1–9, 2014.

[18] A. Azfar, K.-K. R. Choo, and L. Liu, "A study of ten popular Android mobile VoIP applications: are the communications encrypted?" in Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS '14), pp. 4858–4867, IEEE, Waikoloa, Hawaii, USA, January 2014.

[19] A. Azfar, K. K. R. Choo, and L. Liu, "Forensic taxonomy of popular Android mHealth apps," in Proceedings of the 21st Americas Conference on Information Systems (AMCIS '15), San Juan, Puerto Rico, August 2015.

[20] A. Azfar, K. K. R. Choo, and L. Liu, "An android communication app forensic taxonomy," Journal of Forensic Sciences, vol. 61, no. 5, pp. 1337–1350, 2016.

APPENDICES

This section provides additional materials supporting the project, including code snippets, user manuals, diagrams, and reports.

Appendix A: Code Snippets

This appendix contains essential code snippets from the project, demonstrating the implementation of key security features. The following are brief excerpts from the project's security tools:

Website Passive Scanner (Security Header Check)

```
import 'package:http/http.dart' as http;

Future<void> checkSecurityHeaders(String url) async {
  final securityHeaders = [
    "X-Frame-Options",
    "X-XSS-Protection",
    "Content-Security-Policy",
    "Strict-Transport-Security"
  ];
  try {
    final response = await http.get(Uri.parse(url));
    final headers = response.headers;

    for (final header in securityHeaders) {
      if (!headers.containsKey(header.toLowerCase())) {
        print("Missing security header: $header");
      } else {
        print("$header is properly set");
      }
    }
  } catch (e) {
    print("Error checking headers: $e");
  }
}
```

DHCP Analyzer (Detecting Rogue Devices)

Directly performing low-level network packet sniffing like the scapy example is not possible in standard Flutter because it is a client-side framework. Network analysis is an operating system-level task that requires elevated privileges and is typically handled by a dedicated backend service or a native application. A Flutter app would act as a user interface that receives data from this backend service

SQL Injection Prevention (Using Parameterized Queries in Python)

```
import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart';


Future<List<Map<String, dynamic>>> secureQuery(String userInput) async {
  final databasePath = await getDatabasesPath();
  final path = join(databasePath, 'database.db');


  final database = await openDatabase(
    path,
    version: 1,
    onCreate: (db, version) {
      return db.execute(
        "CREATE TABLE users(id INTEGER PRIMARY KEY, username TEXT)",
      );
    },
  );


  final query = "SELECT * FROM users WHERE username = ?";
  final result = await database.rawQuery(query, [userInput]);


  await database.close();
  return result;
}
```

These code snippets highlight key components of the project's implementation, ensuring security vulnerabilities are properly mitigated.

Appendix B: User Manual

This section provides guidelines for users to interact with the system effectively.

Launching the Vulnerability Scanner

To use the Vulnerability Scanner feature in a Flutter application, users would follow these simple steps:

1. **Launch the App**: Open the installed application on your device by tapping its icon.

2. **Enter URL**: Find the input field on the main screen, enter the full website URL (e.g., https://example.com), and press the "Scan" button.

3. **View Results**: The application will then display the scan results directly on the screen, showing security headers and potential vulnerabilities.

Using the DHCP Analyzer

- Ensure the system has appropriate privileges to monitor network traffic and Run the following command: python dhcp_analyzer.py
- The tool will automatically detect and alert about unauthorized devices attempting to join the network.
- **Reviewing Security Logs**
  - Security logs are stored in the logs/ directory within the project folder.
  - Open the log files using a text editor to review detected threats and security alerts.

This manual serves as a quick reference for users interacting with the security tools.

Appendix C: Diagrams

System Architecture

User → Application/Web/Software → Security Framework → Threat Detection → Logging & Alerts

Illustrates how the security framework integrates into a web application to detect and mitigate threats.