Final Year Project Proposal

Development of Security Assest Classification & Analysis Tool

Supervisor

Mr. Mohammad

Submitted by

Palwasha Javed

21PWBCS0840

Ayesha Saleem

21PWBCS0844

Maimona Marjan

21PWBCS055

Department of Computer Science & Information Technology,

KPK University of Engineering & Technology, Peshawar

Date of Submission

18/10/2024

Contents

1. Introduction

In today's digital age, web-based systems play a critical role in business and government operations, processing sensitive data like personal information, financial records, and confidential documents. However, these systems are increasingly vulnerable to cyberattacks , which threaten data security. To address this issue, my final year project will focus on developing a framework for systematically classifying assets for security testing based on their exposure, confidentiality, integrity, and availability (CIA) needs. To improve security, the framework will categorize systems into High, Medium, or Low priority for testing. This classification is based on calculated parameters such as Exposure Factor (EF), Cumulative CIA (CCIA) score, and Asset Value Level (Ac). By prioritizing security testing based on these classifications, we can ensure that resources are allocated to secure the most vulnerable and critical systems first, reducing the risk of data breaches. Additionally, the project will implement a keyword-based questioning system that automatically assesses the security status of systems using Natural Language Processing (NLP) to reduce false positives. By combining these tools, the framework aims to enhance the security landscape of web systems across various sectors.

2. Aim and Objective of Proposal

The primary aim of this project is to develop a robust asset security classification framework for prioritizing security testing in web-based systems. This framework will categorize systems into High, Medium, or Low priority based on factors such as exposure to external threats, the sensitivity of the data processed, and the overall value of the asset to the organization. Using this classification, the goal is to ensure that high-risk systems receive more focused security testing.

Specific Objectives:

- **Classifying Assets by Their Security Needs:**
  Systems will be grouped into High, Medium, or Low priority for security testing, based on their calculated Exposure Factor (EF), Cumulative CIA (Confidentiality, Integrity, Availability) score, and Asset Value Level (Ac). This classification will help in prioritizing which systems need immediate security attention.
- **Vulnerability Testing Using OWASP Guidelines**:
  Security testing will be conducted according to OWASP guidelines, targeting common vulnerabilities such as SQL injection, broken access control, and cross-site scripting (XSS) to identify and mitigate weaknesses in web-based systems.
- **Implementing a Keyword-Based Questioning System**:
  This objective focuses on creating a keyword-based questioning system to assess the security status of systems, using NLP to minimize false results due to keyword confusion.
- **Checking CIA Metrics via Third-Party APIs:**
  This objective will check the Confidentiality, Integrity, and Availability (CIA) metrics as percentages by leveraging third-party APIs for greater accuracy.
- **Testing the Framework on Real-World Systems:**
  This objective involves evaluating the framework's effectiveness and reliability by testing it on a sample of real-world web-based systems.
- **Providing Recommendations for Security Measures**:
  Based on the testing results, this objective focuses on offering recommendations for security measures and testing tools tailored to the classification of each system.

3. Background of the Project Proposal

The rapid proliferation of web-based systems has been met with a corresponding rise in cyber threats , including vulnerabilities such as broken access control, SQL injection, and cross-site scripting (XSS). Security testing is a crucial defense, but organizations often struggle to determine which systems should be prioritized for testing. Inspired by an asset security classification method outlined in a study conducted on 451 web-based systems in Khyber Pakhtunkhwa , Pakistan, this project aims to build a more scalable and adaptable solution. The framework will classify systems into High, Medium, or Low priority for security testing. This classification is based on key metrics such as the system's Exposure Factor (EF), Cumulative CIA (CCIA) score, and Asset Value Level (Ac). Additionally, a keyword-based questioning system, enhanced by NLP, will provide an automated assessment of a system’s security status, while minimizing errors caused by keyword confusion.

4. Learning Outcomes

Upon completion of this project, the following learning outcomes are expected:

- **Understanding Security Principles:** Develop a comprehensive understanding of the CIA triad (Confidentiality, Integrity, Availability) and its application in web-based system security.
- **Security Testing Methodologies:** Learn to create and implement a framework for prioritizing security testing, applying OWASP guidelines for vulnerability testing and risk assessment.
- **Vulnerability Assessment Skills:** Gain proficiency in identifying and addressing common vulnerabilities in web-based systems, enhancing skills in programming and security testing methodologies.
- **Data Analysis and Risk Management:** Acquire the ability to collect and analyze data related to system exposure and security risks using various web scraping and data analysis tools, and develop strategies for effective risk mitigation.

5. Required Skills of the Team Members

To successfully complete the project, the team must possess a range of technical, analytical, and design skills. These are detailed below:

- **Program Design / Programming**: Proficient in **Python** and **JavaScript**, with the ability to design and implement scripts for web scraping, security testing, and automating the classification framework.
- **Natural Language Processing:** Knowledge in NLP algorithms to reduce confusion in keyword-based questioning.
- **Data Analysis / Data Modeling**: Skills in **data analysis tools like Pandas** for processing and interpreting large volumes of security-related data. Ability to apply statistical models and create data visualizations using tools like **Seaborn/Matplotlib**.
- **Database Design / Database Construction:** Proficiency in designing and managing relational databases (e.g., MySQL, PostgreSQL) or non-relational databases (e.g., MongoDB) to store data collected from security tests and analysis. Team members must be able to create structured schemas for efficiently managing asset classification data.
- **Proposal Writing (Formal, Semi-formal, Structured)**: Strong skills in technical writing, necessary for drafting a clear, structured **project proposal**, including sections like background, methodology, and expected outcomes.

- **Program Testing / Formal Verification**: Experience in **formal testing methodologies**, particularly with security testing tools such as **OWASP ZAP or Burp Suite**. Ability to carry out **formal verification** of the testing results, ensuring accuracy and completeness of the vulnerability assessment.
- **User Interface Design / Interface Programming**: Knowledge of **UI/UX design principles** to develop intuitive, functional interfaces for interacting with the security framework or presenting testing results. Experience in **HTML, CSS, and JavaScript** for creating interactive front-end components of the security classification system.
- **Requirements Elicitation / Analysis**: Ability to gather and analyze project requirements by studying real-world web-based systems and identifying their security needs. Experience in translating these requirements into system features that align with security objectives.
- **Experimental Design / Results Analysis**: Skills in designing **experiments** to evaluate the framework's effectiveness on real-world web-based systems. Proficiency in **results analysis**, using metrics like **Exposure Factor (EF)** and **Cumulative CIA (CCIA)** to determine the effectiveness of the classification.
- **Simulation / Emulation / Animation**: While not central to this project, basic understanding of **simulation techniques** (such as mimicking security environments) could help in creating test environments for running security evaluations.
- **Hardware Design**: Though this project is largely software-based, some understanding of **server configurations and hardware architecture** is needed to run security tests and host the web-based systems.

6. Methodology

The project will be approached through the following steps:

- **Information Gathering**: Collect data on the systems under test (SUT) including exposure, user access, and the type of information processed.
- **Security Classification**: Systems will be classified into High, Medium, or Low priority based on their calculated Exposure Factor (EF), Cumulative CIA (CCIA) score, and Asset Value Level (Ac). High-priority systems will undergo more rigorous testing.
- **Keyword-Based Security Checks**: A keyword-based questioning system will automatically assess security by analyzing keyword matches in system responses. If three or more keywords match as "false," the system will be flagged as insecure. NLP will be used to minimize confusion between similar terms in different contexts.

- **OWASP Vulnerability Testing**: High-priority systems will be subjected to comprehensive vulnerability testing based on OWASP's Top 10 vulnerabilities to ensure accuracy and effectiveness.
- **Analysis of Results**: The results of the security classification and testing will be analyzed to determine how accurately the framework prioritizes security testing.
- **Final Recommendations**: Recommendations for improving security measures will be made based on the results of the framework's implementation, tailored to the priority level of each system.

7. Deliverables/Scope

This project will provide a systematic framework for prioritizing security testing in web-based applications, which is highly relevant for industries that handle sensitive data, such as:

1. **Healthcare:** Protecting patient data and medical records.
2. **Financial Services:** Safeguarding banking and financial information.
3. **E-commerce:** Ensuring customer data protection in online transactions.
4. **Government Websites:** Securing sensitive governmental information and protecting citizen data.

The target domains will benefit from a more efficient use of security testing resources, allowing organizations to focus on the most critical and vulnerable systems.

8. Tools/Technology

The project will utilize a combination of software, hardware, and web technologies to develop the security classification framework and conduct vulnerability testing. The following tools and technologies will be required:

Hardware Tools:

- **Laptop/Desktop**: Required for running simulations, web scraping tools, and performing security testing.
- **Server Infrastructure**: A cloud-based or local server environment to host the web-based systems being tested.

Software Tools:

Programming Languages:

- **Python**: For developing scripts to automate web scraping, data collection, and for framework implementation.
- **JavaScript**: For web application testing and interaction with front-end systems.

Web Scraping & Automation:

- **BeautifulSoup & Selenium**: To scrape relevant data from web-based systems for analysis and classification.

NATURAL LANGUAGE PROCESSING (NLP) TOOLS:

- **Spacy or NLTK:** These NLP libraries will be employed to ensure confusion-free keyword analysis during the security status assessment. They will help distinguish between similar terms in different contexts and reduce false positives in the keyword-based questioning system.

Security Testing Tools:

- **OWASP ZAP or Burp Suite**: To perform vulnerability testing according to OWASP guidelines, identifying weaknesses such as SQL injection, XSS, and broken access control

Data Analysis & Visualization:

- **Pandas**: To manage and analyze data collected from vulnerability testing.
- **Matplotlib/Seaborn**: For visualization of security risks, trends, and asset classifications.

Database Management:

- **MySQL/PostgreSQL**: For storing and managing data collected during the testing and classification processes.
- **MongoDB**: For flexible data storage, allowing easy manipulation of unstructured web scraping data.

Version Control:

- **Git/GitHub**: For managing project code and collaborating across team members.

Development Environment:

- **VS Code/PyCharm**: For writing, debugging, and testing code.

Web-Based Testing Platform:

- A testing platform for running security tests on real-world systems hosted on servers, potentially utilizing a cloud service like AWS or Azure.

9. Expertise of the Team Members

Each team member has the following relevant expertise:

- **Programming**: Team members have completed relevant courses in software development and security testing.
- **Web Development**: Knowledge in web application development is strong across the team, with particular experience in both front-end and back-end technologies.
- **Cybersecurity**: Team members have taken courses in cybersecurity and are familiar with penetration testing techniques and vulnerability assessments.
- **Data Analysis**: We have studied data analysis techniques and will use these skills to evaluate security metrics.
- **NLP:** NLP algorithms to reduce confusion in keyword-based questioning.

The project is equally interesting to all team members, and we are confident in our collective ability to complete the tasks required for successful project execution.

10. Project Timeline

Proposed Started Date	10 Oct 2024
Expected Ended Date	25 JUNE 2025

Activity	2024			2025					
	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN
Literature review									
Framework Design									
Development Phase									
Integration									

Testing									
Final Testing And Development									
Deployment									
Thesis writing									

11. References

- OWASP Foundation. "OWASP Top Ten - 2021." [Online] Available: https://owasp.org/www-project-top-ten/.
- NIST. "Security and Privacy Controls for Information Systems and Organizations." [Online] Available: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final.
- Herzog, P. "OSSTMM 3: The Open Source Security Testing Methodology Manual - Contemporary Security Testing and Analysis." ISECOM, 2010. [Online]. Available: https://www.isecom.org/OSSTMM.3.pdf.
- Di Lucca, G. A., Fasolino, A. R., Faralli, F., & De Carlini, U. "Testing Web Applications." In Proc. Int. Conf. on Software Maintenance, Montreal, Quebec, Canada, 2002.
- Li, X., & Xue, Y. "A Survey on Web Application Security." Nashville, TN, USA: Vanderbilt University, Technical Report, 2011.
- Jan, S., et al. "A Framework for Systematic Classification of Assets for Security Testing." Computers, Materials & Continua, 2021.
- Open Web Application Security Project (OWASP). "OWASP Top 10 Vulnerabilities." OWASP, 2017.
- OWASP ZAP. "OWASP Security Testing Tool." Available: https://www.owasp.org.
- Patel, S., et al. "Risk Assessment Modeling Technique for Cybersecurity." Journal of Industrial Systems, 2018. Available: https://www.isis.vanderbilt.edu/sites/default/files/main_0.pdf.