Software Verification Report

for

Cryptographically - Secure Password Manager

Version 1.0

Prepared by

Anushka Singh | 2021IMG-014
Avijeet Jain | 2021IMG-018
Yuvraj Kumar | 2021IMG-066

ABV-Indian Institute of Information Technology & Management, Gwalior

18th April 2022

Submitted in partial fulfillment
Of the requirements of
IMIT-2202 Software Engineering Lab

Table of Contents

1. Introduction

A cryptographically secured password manager provides a secure and convenient way for users to store and manage their passwords and other sensitive information. A password manager typically holds login credentials for various websites and applications and can also store sensitive information such as credit card numbers, social security numbers, and other personal information.

A cryptographically secured password manager uses robust encryption algorithms to protect the stored data, making it difficult for unauthorized users to access or steal the information. The encryption key used to secure the data is typically derived from a master password chosen by the user, which is stored using robust encryption techniques.

Using a password manager can also make it easier for users to create and manage complex and unique passwords for different accounts, reducing the risk of a password breach. With a password manager, users only need to remember one master password to access all their stored credentials, rather than having to remember multiple passwords for different accounts.

A cryptographically secured password manager can help users protect sensitive information and maintain better security practices.

The purpose of this document is to verify the project PassMan - A Cryptographically secured password manager. Verification is a critical aspect of software development that involves testing and validating the various components and functionalities of a software system to ensure that it meets the required specifications and performance standards.

2. Functional Verification

Functional verification involves testing the system's various functionalities to ensure that they are working as intended and meeting the user's needs. This includes testing the password generation and management module, encryption and decryption algorithms, user authentication module, and database management module. The verification should ensure that the system can securely generate, store, and retrieve passwords, as well as authenticate users and protect sensitive data.

In this project, the functional verification process will include testing the password generation and management module, encryption and decryption algorithms, user authentication module, and database management module.

2.1 Password generation and management module

This module should be tested to ensure that it can securely generate and store passwords, as well as retrieve them when needed. Testing should include creating, updating, and deleting passwords, as well as testing the password strength and complexity algorithms.

2.2 Encryption and decryption algorithms

The encryption and decryption algorithms should be tested to ensure that they are robust and effective in protecting user data. Testing should include encrypting and decrypting data using different encryption algorithms and key lengths, as well as testing the system's resistance to various attack vectors, such as brute force attacks and dictionary attacks.

2.3 User authentication module

The user authentication module should be tested to ensure that it can securely authenticate users and protect sensitive data. Testing should include creating, updating, and deleting user accounts, as well as testing the system's resistance to various authentication attacks, such as password guessing and credential stuffing.

2.4 Database management module

The database management module should be tested to ensure that it can securely store and retrieve user data. Testing should include creating, updating, and deleting database records, as well as testing the system's resistance to various database attacks, such as SQL injection and cross-site scripting.

3. Performance Verification

Performance verification involves testing the system's performance and speed to ensure that it is meeting the required performance standards. This includes testing the system's response time, processing speed, and overall system efficiency. The verification should ensure that the system can efficiently manage and process large amounts of data without experiencing any significant delays or performance issues.

In this project, the performance verification process will include testing the system's response time, processing speed, and overall system efficiency.

3.1 Response time

The response time of the system should be tested to ensure that it is fast and efficient. Testing should include measuring the time it takes for the system to perform various tasks, such as generating and retrieving passwords, encrypting and decrypting data, and authenticating users.

3.2 Processing speed

The processing speed of the system should be tested to ensure that it can efficiently manage and process large amounts of data. Testing should include stress testing the system by simulating high user loads and testing the system's response time and processing speed under these conditions.

3.3 Overall system efficiency

The overall system efficiency should be tested to ensure that it is meeting the required performance standards. Testing should include measuring the system's memory usage, CPU usage, and other performance metrics to ensure that the system is running efficiently and without any significant performance issues.

4. Security Verification

Security verification involves testing the system's security measures and protocols to ensure that they are robust and effective in protecting user data from unauthorized access or hacking. This includes testing the encryption and decryption algorithms, user authentication module, and database management module for any potential security vulnerabilities. The verification should ensure that the system can securely protect user data and prevent any potential security breaches.

Security verification involves testing the system's security measures and protocols to ensure that they are robust and effective in protecting user data from unauthorized access or hacking. In this project, the security verification process will include testing the encryption and decryption algorithms, user authentication module, and database management module for any potential security vulnerabilities.

4.1 Encryption and decryption algorithms

The encryption and decryption algorithms should be tested for any potential security vulnerabilities, such as key weaknesses, side-channel attacks, or other cryptographic weaknesses. Testing should include simulating various attack scenarios to ensure that the system can withstand potential attacks and prevent any data breaches.

4.2 User authentication module

The user authentication module should be tested for any potential security vulnerabilities, such as weak passwords, password reuse, or other authentication weaknesses. Testing should include simulating various attack scenarios, such as password guessing or brute force attacks, to ensure that the system can securely authenticate users and protect sensitive data.

4.3 Database management module

The database management module should be tested for any potential security vulnerabilities, such as SQL injection, cross-site scripting, or other database weaknesses. Testing should include simulating various attack scenarios to ensure that the system can securely store and retrieve user data without any security breaches.

5. Usability Verification

Usability verification involves testing the system's user interface and user experience to ensure that it is intuitive, user-friendly, and easy to navigate. This includes testing the system's user interface design, functionality, and accessibility. The verification should ensure that the system is easy to use and understand, even for users who may not have extensive technical knowledge or experience.

- Usability verification aims to ensure that the password manager is user-friendly and easy to use. Usability verification was carried out by conducting user testing on a group of participants who were not involved in the development of the password manager. The goal was to observe how users interacted with the system, and identify any issues or areas for improvement.

- During the usability testing, participants were given a series of tasks to perform using the password manager, such as creating a new password, adding a new account, or searching for a password. Participants were also asked to provide feedback on their experience using the system.

- The results of the usability testing were analyzed to identify any areas where the password manager could be improved. Some of the issues that were identified included confusing terminology, unclear instructions, and difficulties navigating the system.

- Based on the feedback from the usability testing, the development team made several changes to the password manager to improve its usability. These changes included updating the terminology used in the system to make it more user-friendly, providing clearer instructions, and simplifying the navigation.

Overall, the usability testing was successful in identifying areas where the password manager could be improved, and the changes made to the system resulted in a more user-friendly and easy-to-use interface.

6. Compatibility Verification

Compatibility verification involves testing the system's compatibility with different devices, platforms, and software applications. This includes testing the system's compatibility with different operating systems, web browsers, and mobile devices. The verification should ensure that the system is compatible with a wide range of devices and platforms, and can seamlessly integrate with other software applications or services.

- Compatibility verification ensures that the password manager is compatible with different operating systems, web browsers, and hardware configurations. The password manager was tested on a variety of platforms and devices to ensure that it works correctly and reliably across different environments.

- The testing was conducted using a range of operating systems, including Windows, MacOS, and Linux, as well as different web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge. The password manager was also tested on different hardware configurations, including laptops, desktop computers, and mobile devices.

- The testing involved installing the password manager on each platform or device and performing a series of tasks, such as creating a new password, adding a new account, or searching for a password. The system was also tested for compatibility with different hardware configurations, such as different screen sizes and resolutions.

- During the compatibility testing, any issues or errors that arose were documented and addressed by the development team. For example, issues were identified with certain versions of web browsers and operating systems, and these were resolved through software updates and patches.

The compatibility testing was successful in identifying any issues with the password manager's performance on different platforms and devices, and the development team was able to make the necessary changes to ensure that the system works correctly and reliably across a range of environments.

Conclusion

Overall, the verification report for the Cryptographically Secured Password Manager project should ensure that the system meets all of the required specifications and performance standards, and is robust, secure, and easy to use for users. Any identified issues or bugs should be properly documented and addressed before the system is deployed or released to the users.