**Logo, company name

Description automatically generated**

**Security Assessment Report**

Version N.0

May 1, 2023

Table of Contents

[1. Summary 3](#_Toc127779445)

[1. Assessment Scope 3](#_Toc127779446)

[2. Summary of Findings 3](#_Toc127779447)

[3. Summary of Recommendations 4](#_Toc127779448)

[2. Goals, Findings, and Recommendations 4](#_Toc127779449)

[1. Assessment Goals 4](#_Toc127779450)

[2. Detailed Findings 5](#_Toc127779451)

[3. Recommendations 5](#_Toc127779452)

[3. Methodology for the Security Control Assessment 5](#_Toc127779453)

[4. Figures and Code 7](#_Toc127779454)

[4.1.1 Process flow of System (this one just describes the process for requesting) 7](#_Toc127779455)

[4.1.2 Other figure of code 7](#_Toc127779456)

[5. Works Cited 7](#_Toc127779457)

x

# Summary

The goal of the project is to create a password encryptor manager that can securely store and manage user passwords. The user logs in to the program and enter their passwords and a key, after which the program then encrypts the passwords using the key and saves them to a text file.

## Assessment Scope

There are no external tools, software, or Browser that would be needed. The OS used was Windows, and the application does not run on any other OS.

## Summary of Findings

Of the findings discovered during our assessment, 1 were considered High risks, 1 Moderate risk, 2 Low, and 2 Informational risks. The SWOT used for planning the assessment are broken down as shown in Figure 2.

Chart, bar chart

Description automatically generated

Figure 1. Findings by Risk Level

The high risk was that someone could get unauthorized access to the password manager because of the waeak log in system that was implemented. The moderate risk was that users cannot recover passwords if they loses access to the password manager, or forget the log in. The low risks were user errors, for example mistyping a password. Another was the user forgetting which password was for what account, because the program does not have a naming feature for the passwords. The informational risks were if the user gave weak passwords or using common passwords that are easy to guess. The other was if they wrote down the encrypted passwords, because they were harder to remember.

Table

Description automatically generated

Figure 2. SWOT

Explain which issues were used from above SWOT (which are addressed in this assessment).

## Summary of Recommendations

To solve the high risk issue, the best course of action would be to implent a way for the user to be able to create a username and password the first time, and also add 2FA to verify the user. The moderate risk mostly involves the recovery of the account, and the backing up of data, both of which are essential to a proper program. The low risks could be solved by adding a method to delete passwords that are in the text file through the program, and also create a way of labeling which passwords are for what account or service. The informational risks are tricky to fix, but a solution could be to not accept the passwords entered by the user if they do not meet certain criteria, such as having a special character.

# Goals, Findings, and Recommendations

## Assessment Goals

The purpose of this assessment was to do the following:

* Test the security that my program has and how it can be improved
* See the progress that has been made throughout the semester
* Determine whether the application was ready for release

## Detailed Findings

|  |  |
| --- | --- |
| **Type** | **Explanation** |
| Threat | This is not a unique service and is a competitive field. Many other services do what the application does, and in a much better way. |
| Threat | Having the passwords on the device is a huge threat because anyone with access to the device could get access to the application and hence the passwords. |
| Threat | If a user forgets the login, has a question, or even wants to offer feedback, there is no way to get in contact with anyone who could help them. It does not help build trust with users. |
| Weakness | The program only encrypts the passwords with keys entered by users, but the user would still have to type in the original password into wherever they are trying to log in. |
| Weakness | Both methods of encryption used are outdated and could be broken. This makes the passwords that the user entered not as secure as they could be. |
| Vulnerablilites | The user not being able to pick the login credentials make it extremely difficult to remember the information and is risking access to the password manager. |
| Vulnerablilites | Forgetting passwords and login information is extremely common, but there is no way to remover that information within the program. |
| Vulnerablilites | The program accepts weak passwords that could be entered by the user, making even the encrypted versions, not much more secure. |

## Recommendations

|  |  |  |
| --- | --- | --- |
| **Risk Value** | **Ease of Fix** | **Explanation** |
| Informational | Very difficult | To make the application more unique, many more features would need to be added. Enough to make it stand out. |
| Moderate Risk | Moderately difficult | Having online backups, or better encryption of the text files, would help protect the passwords |
| Moderate Risk | No Known Fix | The ability to request assistance would be very helpful, but I am unsure of how to go about implementing it. |
| High Risk | No Known Fix | In order to avoid this issue, the user would have to print the text file, and then manually copy the encrypted password to paste into the proper service. This is incredibly inconvenient. It would be better if once the user enters the password it would automatically copy it to the clipboard. |
| Moderate Risk | Moderately difficult | At the first run time, the user could create a login information and then add 2FA to verify their identity. That way the generic credentials are not being used. |
| Low Risk | Moderately difficult | Using the 2FA from the custom login credentials, the 2FA could also be used to reset the password or log in. |
| Informational | Easy | The checking to see if the user has met all of the criteria would help ensure the strength of the user’s passwords and make them much harder to break. |

# Methodology for the Security Control Assessment

**3.1.1 Risk Level Assessment**

Each Business Risk has been assigned a Risk Level value of High, Moderate, or Low. The rating is, in actuality, an assessment of the priority with which each Business Risk will be viewed. The definitions in Table 1 apply to risk level assessment values (based on probability and severity of risk). While Table 2 describes the estimation values used for a risk’s “ease-of-fix”.

Table 1 - Risk Values

| Rating | Definition of Risk Rating |
| --- | --- |
| High Risk | Exploitation of the technical or procedural vulnerability will cause substantial harm to the business processes. Significant political, financial, and legal damage is likely to result |
| Moderate Risk | Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity and/or availability of the system, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment to organization. |
| Low Risk | Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The confidentiality, integrity and availability of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment |
| Informational | An “Informational” finding, is a risk that has been identified during this assessment which is reassigned to another Major Application (MA) or General Support System (GSS). As these already exist or are handled by a different department, the informational finding will simply be noted as it is not the responsibility of this group to create a Corrective Action Plan. |
| Observations | An observation risk will need to be “watched” as it may arise as a result of various changes raising it to a higher risk category. However, until and unless the change happens it remains a low risk. |

Table 2 - Ease of Fix Definitions

| Rating | Definition of Risk Rating |
| --- | --- |
| Easy | The corrective action(s) can be completed quickly with minimal resources, and without causing disruption to the system or data |
| Moderately Difficult | Remediation efforts will likely cause a noticeable service disruption   * A vendor patch or major configuration change may be required to close the vulnerability * An upgrade to a different version of the software may be required to address the impact severity * The system may require a reconfiguration to mitigate the threat exposure * Corrective action may require construction or significant alterations to the manner in which business is undertaken |
| Very Difficult | The high risk of substantial service disruption makes it impractical to complete the corrective action for mission critical systems without careful scheduling   * An obscure, hard-to-find vendor patch may be required to close the vulnerability * Significant, time-consuming configuration changes may be required to address the threat exposure or impact severity * Corrective action requires major construction or redesign of an entire business process |
| No Known Fix | No known solution to the problem currently exists. The Risk may require the Business Owner to:   * Discontinue use of the software or protocol * Isolate the information system within the enterprise, thereby eliminating reliance on the system   In some cases, the vulnerability is due to a design-level flaw that cannot be resolved through the application of vendor patches or the reconfiguration of the system. If the system is critical and must be used to support on-going business functions, no less than quarterly monitoring shall be conducted by the Business Owner, and reviewed by IS Management, to validate that security incidents have not occurred |

**3.1.2 Tests and Analyses**

This was completed using <list and describe any type of testing you performed here>.

**3.1.3 Tools**

This was completed using <list and describe any tools used for testing (include Linux Command Line commands>.

# Figures and Code

Insert any pictures here (including of major code issues or code that was used as a tool – can just screenshot and add link to github). This section must include at least 4 figures or code portions:

### Process or Data flow of System (this one just describes the process for requesting), use-cases, security checklist, graphs, etc.

Diagram

Description automatically generated

Describe the process flow here.

### Other figure of code

HERE

# Works Cited

**There are no sources in the current document.**