Security Assessment Report

Joshua Wurtenberg

Professor Greenwell

CEN 3078

May 1, 2023

Table of Contents

1. Summary 3
    1. Assessment Scope 3
    2. Summary of Findings 3
    3. Summary of Recommendations 4
2. Goals, Findings, and Recommendations 4
    1. Assessment Goals 4
    2. Detailed Findings 5
    3. Recommendations 5
3. Methodology for the Security Control Assessment 5
4. Figures and Code 7
        4.1.1 Process flow of System (this one just describes the process for requesting) 7
        4.1.2 Other figure of code 7
5. Works Cited 7

1. Summary

The overall goal of the original project was to create a mock database with a GUI and auditing. Therefore, I wanted to increase the security of the program with topics learned throughout this course. One of the most notable ways I did this was by adding encryption to all the plaintext files holding critical information about “employees”.

1. Assessment Scope

The only OS that was tested was windows, since I was unable to get Qt working effectively on Mac OS or Linux. With that being said, Qt was tested heavily during the entirety of this project. Not only did I ensure the version I used was up-to-date, but I also spent so much time debugging my program and researching QT that if there were any issues with the library, they would have presented themselves.

Additionally, I tested my “encryption” and “decryption” methods many different times, under many different conditions, and found no problems. A specific method of testing that I used was giving my laptop to my friend and seeing if he could find any holes in the program.

2. Summary of Findings

Unfortunately, I found a lot of severe issues. However these issues vary on how likely they are to happen, and most of them are easily fixable.

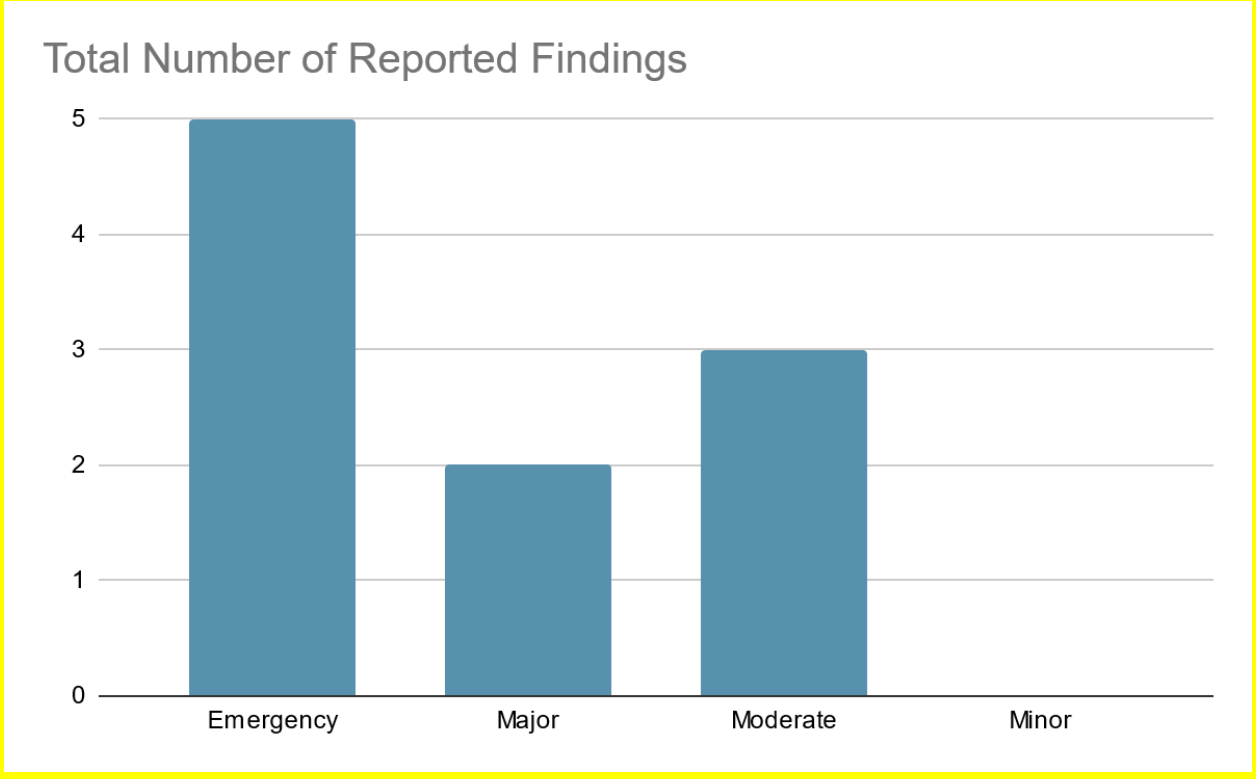

Figure 1. Table of Risks

Strength	Weaknesses	Opportunities	Threats
To prevent the loss of all data in the event of my laptop being stolen, having cloud backups or physical hard drives would be extremely helpful.	Lack of 2 factor authentification.	Because the program and data are only on one laptop, it makes security a lot simpler and easier to implement.	The files being on one laptop could lead to theft of the laptop, causing confidential employee data to be leaked.
In order to make the passwords and data more secure, I could create an algorithm in the program to encrypt and decrypt data before reading and writing it.	Passwords are not encrypted.	Since all the logic in the program was created by me, it makes debugging and adding features a lot easier.	Clicking a link on an untrustworthy email could lead to an employee being the subject of a phishing attack, causing hackers to have access to the unencrypted files.
	Confidential employee data is kept on a desktop text file.		Downloading a file that looks trustable, but is not could lead to somebody having access to the public desktop text files causing confidential employee data to be viewed, stolen, and altered.

Figure 2. SWOT diagram

The most notable issue and solution from my SWOT analysis that I addressed in my program was encryption. I realized that it was the first order of business, seeing how all of the information in the program was read and written from a plaintext file. Therefore, I devoted my attention to solving that problem first and foremost.

3. Summary of Recommendations

A lot of things about my program were tweaked for the sake of better security. I uploaded my program to GitHub, giving it the added protection of backups through the use of commits. In addition, the encryption that I added was not only added to the employee information, but also the auditing file. Therefore, all the changes to the database can be monitored and only by those with the encryption key. The only security feature left to be desired is 2FA. I was limited on time and resources, so this was unable to happen. However, when I revisit this project in the future, that will be the first change I make.

2. Goals, Findings, and Recommendations

1. Assessment Goals

The purpose of this assessment was to do the following:

- Ensure protection of sensitive data stored by my program
- learn better computer security practices

2. Detailed Findings

Vulnerabilities -

- User leaving a password on a sticky note. According to comparitech.com, a concerning 42% of businesses rely on sticky notes to remember passwords. This is extremely dangerous, because almost half of businesses could easily succumb to a password being swiped off of a sticky note and their data being breached.
- Lack of 2 factor authentification. Without 2FA, if somebody were to get a hold of one of those sticky notes with employee login information, there would be nothing stopping them from having full access to an employee account and company data.
- Entire program is kept on a laptop. This means that the program and all the information on it could be stolen easily. Additionally, if the computer were to be broken or malfunction somehow, everything would be lost.
- Employee info was kept on a plaintext file. If someone were to have access to my laptop, all they would have to do is open the file and see all the information.
- Auditing was kept on a plaintext file. If someone were to have access to my laptop, they could easily have access to a full history of changes to employee and company data.
- Program was not backed up. If the program were to be infected by a virus, there would be no restore state to return to, and the program would be lost.
- Phishing attack. If I were to be the victim of a phishing attack, they would have access to all the employee information because it is on plaintext files.
- Trojan download. Because the program is on a personal computer, it is much more likely that I download a trojan that affects it.

3. Recommendations

My major fixes come straight out of the SWOT diagram. The items in the strengths category were what I worked on first. I added a backup feature by using GitHub. Therefore, if the program were to be infected, or anything happens to the laptop that houses the program, I would have a means of getting it back. Next I made sure to encrypt all the employee information. Therefore there is no way of reading sensitive data without having access to the system.

My next course of action would be to add a form of 2FA, therefore the data would be even more protected from malicious actors.

3. 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

Rating	Definition of Risk Rating
	• 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

I believe that all of my placements for my assessment were rather accurate. There are no threats that stand out to me as being put in the wrong category.

3.1.2 Tests and Analyses

I found out most issues with my encryption methods through the use of whitebox testing.

3.1.3 Tools

I used visual studio as a tool for debugging when I only needed to test certain parts of my program that did not involve the GUI. This helped me from having to build and run through the entire program just to run through minor issues.

4. Figures and Code

4.1.1 Encryption and Decryption code

```
30  char CODEX[] = { '0', '1', '2','3','4','5','6','7','8','9','a','b','c','d','e','f','g','h',
31  'i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E',
32  'F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','!','#',
33  '$','%','&','(',')','*','+',',',' ','-','.','/',':',';','<','=','>','?','@','[',']', '^',
34  '_','`','{', '|', '}','~' };
35
36  int find(char a[], int len, char find) {
37      for (int i = 0; i < len; i++) {
38          if (a[i] == find) {
39              return i;
40          }
41      }
42      return -1;
43  }
44  std::string encrypt(std::string s, int n) {
45      int index = s.size();
46      int x;
47      for (int i = 0; i < index; i++) {
48          x = find(CODEX, sizeof(CODEX), s[i]);
49          s[i] = CODEX[(x + n) % sizeof(CODEX)];
50      }
51      return s;
52  }
53  std::string decrypt(std::string s, int n) {
54      int index = s.size();
55      int x;
56      for (int i = 0; i < index; i++) {
57          x = find(CODEX, sizeof(CODEX), s[i]);
58          s[i] = CODEX[(sizeof(CODEX)+(x-n)) % sizeof(CODEX)];
59      }
60      return s;
61  }
```

This is the code that contains the codex and methods for encryption and decryption.

4.1.2 My Security Assessment

		Probability ---------->				
Severity		Frequent	Probable	Likely	Possible	Rare
|	Emergency	A user could leave their password on a sticky note, making it even easier to steal confidential information.	Lack of 2 factor authentification.	Downloading a trojan would give access to all the files and program.	Phishing attacks would easily be able to steal confidential informations from a non-encrypted text file	Someone could steal the laptop the program and text files are stored on.
|	Major	Confidential employee data is read and written to and from a desktop text file.	Employee passwords are read and written to and from a desktop text file.			
|	Moderate	Confidential employee information is not encrypted.	Passwords are not encrypted.	Audit logs are not encrypted.		
|	Minor					
|	Negatable					
V						

5. Works Cited

"C And C++ Reference." *Cppreference.com*, https://en.cppreference.com/w/.

"Encryption 101." *EDUCAUSE.edu*, https://www.educause.edu/focus-areas-and-initiatives/policy-and-security/cybersecurity-program/resources/information-security-guide/toolkits/encryption-101.

"The Future Is Written with Qt." *Qt Documentation*, https://doc.qt.io/.

Greenwell, Josiah, director. *SimpleEncryptionPython* , 27 Feb. 23AD, https://web.microsoftstream.com/video/f5afb5ff-7139-464f-b478-37f7ca5051c2. Accessed 1 May 2023.

O'Driscoll, Aimee. "25+ Password Statistics That May Change Your Password Habits." *Comparitech*, 24 Mar. 2023, https://www.comparitech.com/blog/information-security/password-statistics/.

"Tutorials." *Cplusplus.com*, https://cplusplus.com/doc/.