Logo, company name

Description automatically generated

Template based on the Centers for Medicare & Medicaid Services, Information Security & Privacy Management’s Assessment

**Security Assessment Report**

Version N.0

2023-05-01

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)

# Summary

Executive Summary Here: Describe the overall goal, method, and major findings/recommendations here. (it’s the TLDR)

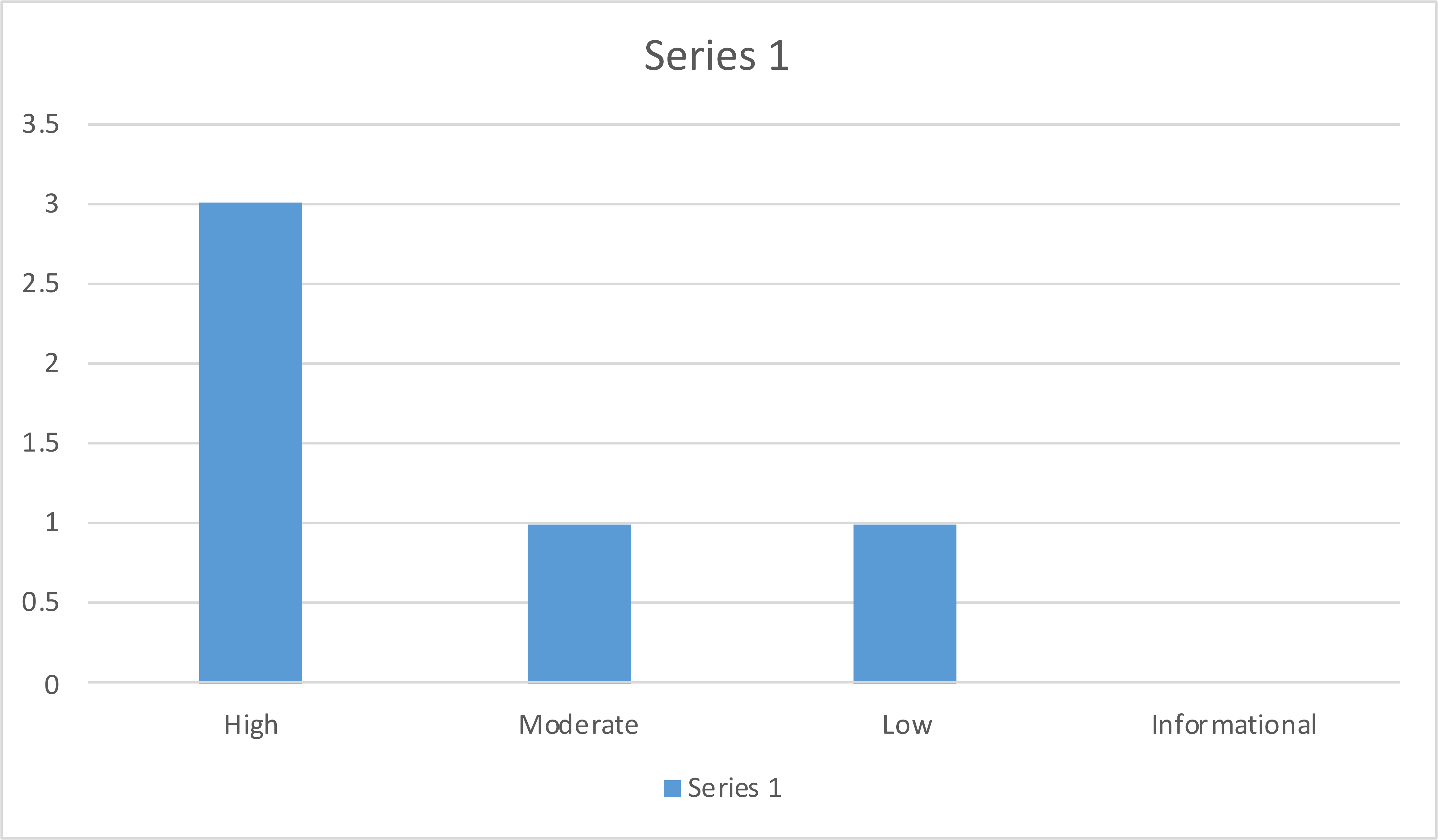
## Assessment Scope

What tools, platforms, OSes, Browsers, and software (including your own) was tested or used in testing?

When I test software, I use a method called manual testing. It means that I do the testing myself and make sure that the software works the way it's supposed to. I start by figuring out what tests need to be done. Next, I set up the software and run the tests. If something doesn't work, I do the test again to make sure it wasn't a mistake.

## Summary of Findings

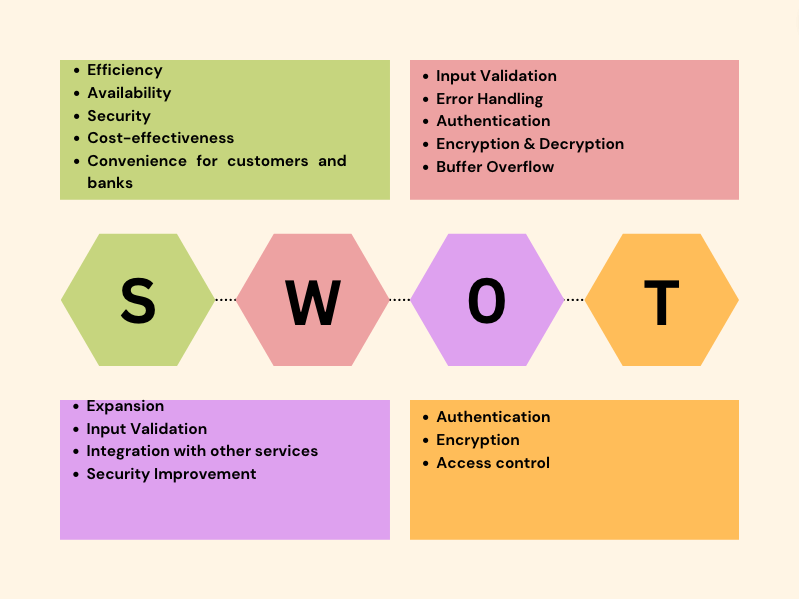
Of the findings discovered during our assessment, 0 were considered High risks, 2 Moderate risks, 0 Low, and 0 Informational risks. The SWOT used for planning the assessment are broken down as shown in Figure 1.



During our assessment several risk was found including: Input Validation, Error Handling, Authentication, Encryption & Decryption, Buffer Overflow, insecure Cryptography storage.

Figure 1. Findings by Risk Level

Explain above and link to full table of explanation of top risks like Figure 2.



Security issues such as input validation, error handling, authentication, encryption and decryption, and buffer overflow are important considerations for ensuring the security and integrity of ATM projects.

Figure 2. SWOT

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

## Summary of Recommendations

To improve the security and reliability of the ATM system, it may be necessary to enhance the validation process for user transactions, implement a more advanced error-detection system to catch rounding errors, improve error handling, and prioritize the implementation of a robust authentication system to protect sensitive data.

# Goals, Findings, and Recommendations

## Assessment Goals

The purpose of this assessment was to do the following:

I believe that the assessment was conducted to evaluate the security and reliability of the ATM project by identifying potential vulnerabilities and risks related to input validation, error handling, authentication, encryption and decryption, buffer overflow, and insecure cryptographic storage. The purpose of the assessment was to identify areas of improvement and implement necessary changes to enhance the security and overall performance of the ATM system.

## Detailed Findings

Ensure each vulnerability is thoroughly explained, specific risks to the continued operations are identified, and the impact of each Threat or Weakness is analyzed as a business case. Ensure these are linked to Table 1 when describing the Risk Value. This is not the fixes – it’s the description of the problems found. The fixes go in the next section (for ease of lookup using TOC) - build this off your checklist, SWOT, and risk assessments.

Broken authentication and session management can lead to unauthorized access and data compromise, insufficient encryption and hashing can result in sensitive data interception and loss of user trust, insecure direct object references can lead to data theft and system downtime, insufficient input validation can lead to system compromise and data theft, buffer overflow can cause system compromise and data theft, insecure cryptographic storage can result in data theft and loss of user trust, and poor error handling can reveal sensitive information and potentially result in data theft and system downtime.

## Recommendations

Here’s where your fixes go (ensure you reference Table 2 for your ease of fix evaluation and explain why it matches that category).

* Validate User Transaction (High, Difficult): This vulnerability requires a significant amount of effort to fix, as it involves extensive code changes and thorough testing to ensure the system's integrity and reliability.
* Rounding Error (High, Easy): This vulnerability is relatively easy to fix with minimal code changes, but it still requires thorough testing to ensure the system's accuracy.
* Improve Error Handling (Low, Easy): This vulnerability is relatively easy to fix with minimal code changes and can be addressed by implementing better error handling mechanisms.
* Authentication (Mid, Moderate): This vulnerability requires a moderate level of effort to fix, as it involves implementing a robust authentication mechanism that ensures the system's security and reliability.
* Encryption & Decryption (High, Difficult): This vulnerability requires a significant amount of effort to fix, as it involves implementing strong encryption and decryption mechanisms that protect sensitive data from unauthorized access or manipulation.

# Methodology for the Security Control Assessment

**3.1.1 Risk Level Assessment (delete this text: you don’t have to change 3.1.1)**

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 Error: Reference source not found 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>.

To ensure the ATM system's functionality, I can perform black box testing. I can analyze the system's code and internal workings to detect potential vulnerabilities using white box testing. Moreover, I can simulate real-world attacks via penetration testing to uncover vulnerabilities and weaknesses in the system.

**3.1.3 Tools**

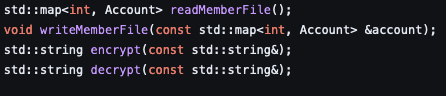
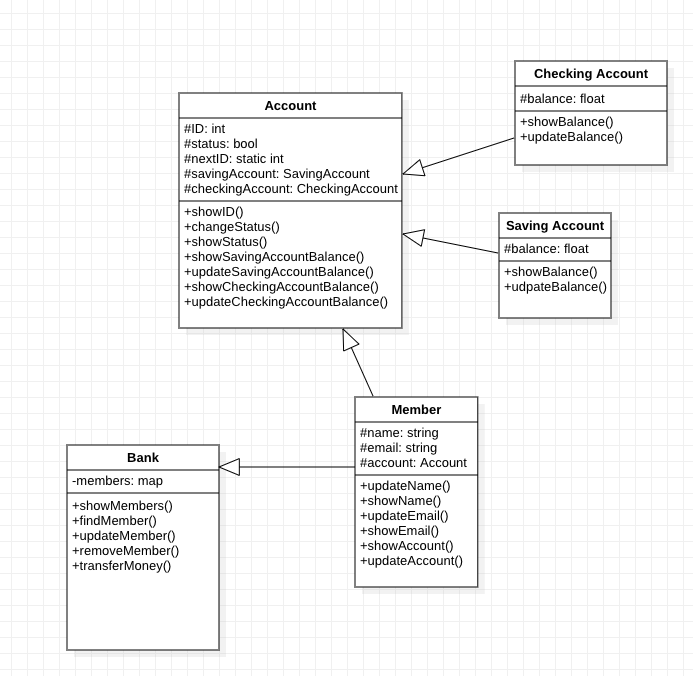
This was completed using <list and describe any tools used for testing (include Linux Command Line commands>.

I can use the terminal to execute test scripts and programs in Unix-based systems. This allows me to interact with the system directly, view system logs and outputs, and run tests with various configurations.

When testing software, I can use a debugger to identify and debug defects. By stepping through the code line by line and examining variables and data structures, I can pinpoint the root cause of the problem. Debuggers are often integrated into IDEs or used as standalone tools.

# 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.

The security assessment process involves planning, scoping, information gathering, analysis, and reporting to identify and address security risks and vulnerabilities.

### Other figure of code

HERE

# Works Cited

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

**OWASP. (n.d.). Broken Authentication and Session Management. Retrieved from** <https://owasp.org/www-community/attacks/Broken_Authentication_and_Session_Management>

·  OWASP. (n.d.). Cryptographic Storage. Retrieved from <https://owasp.org/www-community/attacks/Cryptographic_Storage>

·  OWASP. (n.d.). Insecure Direct Object References. Retrieved from <https://owasp.org/www-community/attacks/Insecure_Direct_Object_References>

·  OWASP. (n.d.). Injection. Retrieved from <https://owasp.org/www-community/attacks/Injection>

·  OWASP. (n.d.). Buffer Overflow. Retrieved from <https://owasp.org/www-community/attacks/Buffer_overflow_attack>

·  OWASP. (n.d.). Insecure Cryptographic Storage. Retrieved from https://owasp.org/www-community/attacks/

https://www.learncpp.com/cpp-tutorial/stdcin-and-handling-invalid-input/

<https://www.geeksforgeeks.org/encrypt-and-decrypt-text-file-using-cpp/>