Security Assessment Report

Version N.1

April 25, 2023

Andrew A. Krupp

Table of Contents

1. Summary

The goals of this project is to assess and fix all security risks found in a project from COP 3003 named: ShapeCalculator. Findings mostly surround the planning of the project. From the beginning, security was not a concern in the ceation of this project.

1. Assessment Scope

This project was built for Windows OS and Windows OS exclusively. To test this code, the program was given to users that have very little technology experience/literacy and given no direction. Another testing strategy was to use Red Hat tactics to gain access to sensitive data or to crash the computer being used (Code Injection Lecture).

Major limitations of the process were caused by the limitations of the resources available to me as a University student. These limited resources (including money, time, knowledge, etc) caused a limited testing process, which means many of the possible security risks were most likely missed.

2. Summary of Findings

Of the findings discovered during our assessment, 2 were considered High risks, 5 Moderate risks, 6 Low, and 0 were not a risk. The SWOT used for planning the assessment are broken down as shown in Figure 1.

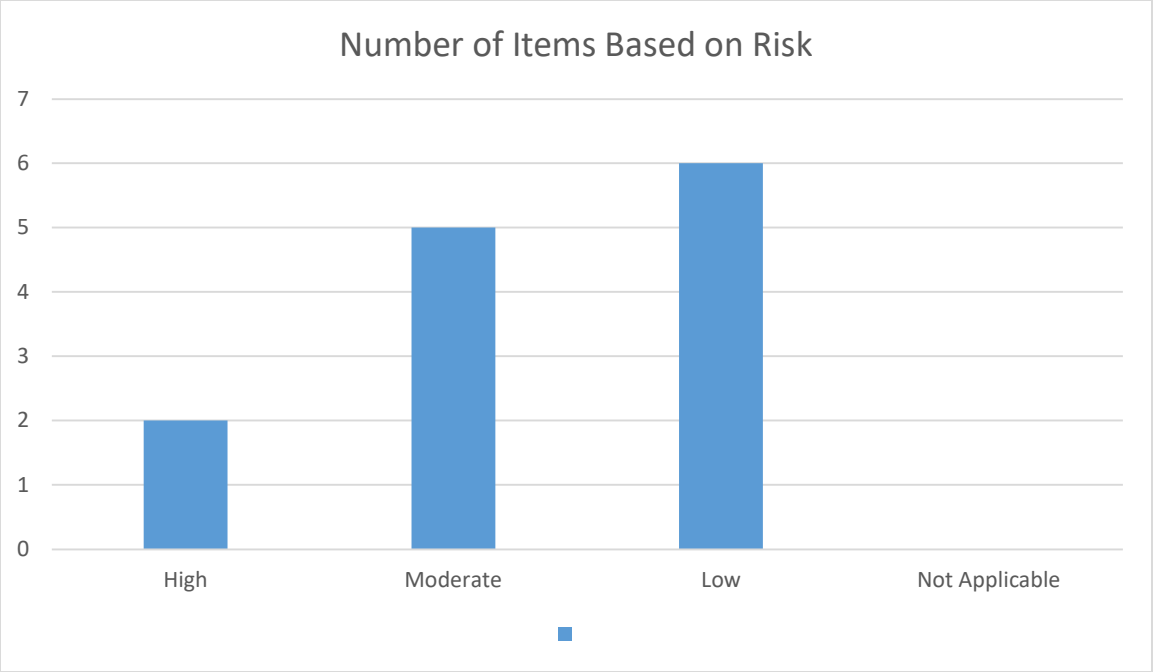


Figure 1. Findings by Risk Level

These values come from Table 1.

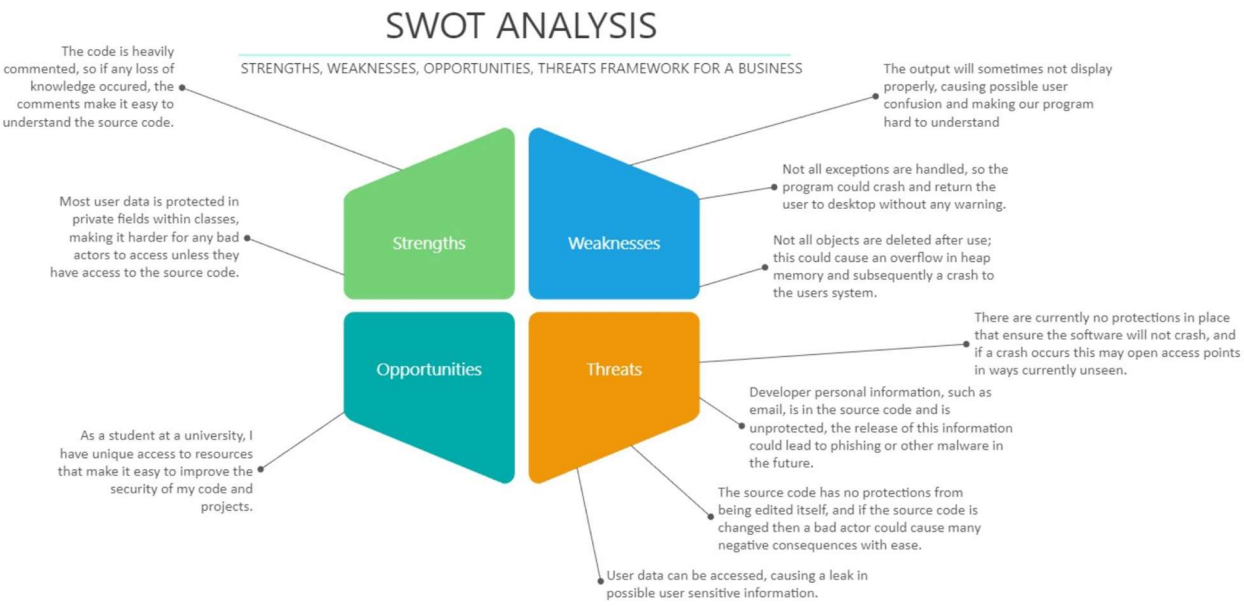


Figure 2. SWOT

In Figure 2, the SWOT analysis has listed weaknesses and threats were converted into some of the security findings in Table 1. For example, the weakness regarding the exception handling turned into finding 8. The threat regarding the access to the source code was turned into finding 1.

3. Summary of Recommendations

It is this documents recommendation that this project be rebuilt from the ground up to consider security risks from the very beginning. This is because this project was built soley to fulfill requirements for a university class, with no consideration of security. This has caused security risks to be baked into the very structure of this codebase.

2. Goals, Findings, and Recommendations

1. Assessment Goals

The purpose of this assessment was to do the following:

- Ensure that the system was in compliance with regulations you had to deal with or any other requirements (to include the assignments themselves).
- Determine if the application was securely maintained.
- Etc.

2. Detailed Findings

Table 1: Security Findings

ID	Issue	Applicable	Complete	Priority	Ease	Description
1	A cloud based platform is being used for access control with only public available items being	Y	Y	Low	Easy	The cloud platform used is Microsoft Teams because it allows multiple people to access multiple files

	readable by general public.					and multiple file types. Public files are put on Github.
2	The cloud based platform provides for hiding information and this is used to protect sensitive information and code.	Y	Y	Low	Easy	Microsoft provides extensive encrypting of cloud hosted files.
3	Buffer Overflow injections are possible using the pointers in the code	Y	Y	Mid	Moderate	Pointers are used in the code to reference places in memory where items created by the code exist. Using pointers is dangerous, but it is what was required by the project
4	Platform user groups are used to only allow changes to be made to code by authorized individuals.	Y	Y	Mid	Easy	All team members are currently part of a private team on microsoft teams and only team members have access to the files.
5	Backup Policy is in place and being used.	Y	Y	Low	Easy	All files are uploaded to github using git and teams manually with their version number and a txt file describing changes.
6	Third-Party libraries used in code are up-to-date and have been checked to ensure no security issues exist.	N	N	Mid	Not Fixable	The programming language used, C++, has an extensive amount of documentation.
7	Physical Security of actual computer code is stored on is adequete	N	N	High	Easy	The computer of the developer is password, pin, or FaceID protected. But the

						password is short and easy to guess. Changing this password to be more robust and requiring two factor authentication would fix this.
8	Accounting: Logging is integrated into the code itself (for exceptions, errors, and user input failures at minimum)	N	N	Low	Moderate	Errors in input have a low probability of being caused by security breaches. The purpose of the code is to visualize data, invalid data input is the main issue, and that is not logged. Adding clauses to the code that add error logging to a txt file would fix this issue.
9	Accounting: Process includes logging (tracking of changes, user making changes, access attempts, etc)	N	N	Mid	Moderate	Minute changes to files are not tracked, only big picture changes. Adding the minute changes being made to the daily txt file would fix this issue.
10	PKI and other encryption and authentication methods are used to connect to cloud platform	Y	Y	Low	Moderate	Microsoft Certificate Authority (CA) is part of the Windows Server operating system. A certification authority (CA) is responsible for attesting to the identity of users, computers, and organizations. The CA authenticates an entity and vouches for that identity by issuing a digitally signed certificate.
11	Internal Actor threats are accounted for and policies/planning	Y	Y	Low	Moderate	The people hired to work on this project is very small and selected

	is in place for these.					specifically for their trustworthiness
12	Standard Unit Testing used	Y	Y	Mid	Moderate	Testing is done periodically through development to ensure that it works with all data provided. To make this better, as noted below, the data should be protected as it is processed in the program
13	Code protects incoming sensitive data from user	N	N	High	Difficult	Data is not protected in any way as it enters and exits code. To protect this, data may need to be encrypted on enter and exit
14	Project was poorly planned	N	N	Low	Difficult	The project was poorly planned from the beginning with no regard for security or the associated risks. Because of this there are many many security risks that could not be fixed without redoing the entire codebase.

3. Recommendations

Based on the simplicity of this project (a simple "shape" calculator created during a semester where hurricane Ian occurred) I struggled to find code-based security issues besides the obvious pointer issues. Because of this, I chose to focus on issues surrounding the implementation of the planning phase or the repository/git usage with the project. Below is the list of fixed or acknowledged issues. In general, exception handling was added, git was used for version control,

and poor planning was addressed. It is this documents recommendation that this project be rebuilt from the ground up to consider security risks from the very beginning.

1.5 When analyzing the github repo description (https://github.com/aakrupp/ShapeCalculator) you can see I focused on the usage of git commands and usage of the git version control abilities. This allowed me to control the versions of the code and to branch where I decided to go a different direction with the code. This control was novel to me, and really enhanced my abilities as an engineer. If, for example, I wanted to share this code base with a partner I could immediately add someone to this repo, set up their git code base on their PC, and they would be set to go. Git would handle any issues with clashing and congruency issues.

1.3 The one code issue I did chose to focus on for this fix was the use of pointers. Using pointers can not only lead to overflow issues, but also data leaks if pointers are not deleted after code finish. To fix this, I have decided to implement SmartPointers to get rid of most of the security issues associated with pointers.

1.14 Project planning was extremely limited for this project. In the future, I would want to implement SmartPointers from the beginning and build the entire code base around that. As of now, the SmartPointers are implemented haphazardly, almost like a patch on an old shirt. With implementation from the beginning, the code base could be more secure and useable overall. Of course, this project was crafter in a week following hurricane Ian, so more planning in any place would be extremely useful in all facets.

1.8 This item was handled in commit 050a41d9ae586ebf4b9c16d5cd610a0cd385452f. I added exception handling for all inputs. This does not handle all possible exceptions that can be handled, but with the time I had I was only able to handle input errors. I handled this using

try and catch blocks anywhere there was an input. This should prevent the code from crashing if the user enters in unexpected values.

1.3 This item was handled in commit 050a41d9ae586ebf4b9c16d5cd610a0cd385452f. I added buffer overflow protection to the arrays existing in the code. It may not be perfect, but it's a start to the problem of buffer overflow issues in this project. I fixed this using the std vector class to use dynamic memory rather than predefined memory. With the vector existing in the heap, the risk of a buffer overflow injection becomes far less of an issue.

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 **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

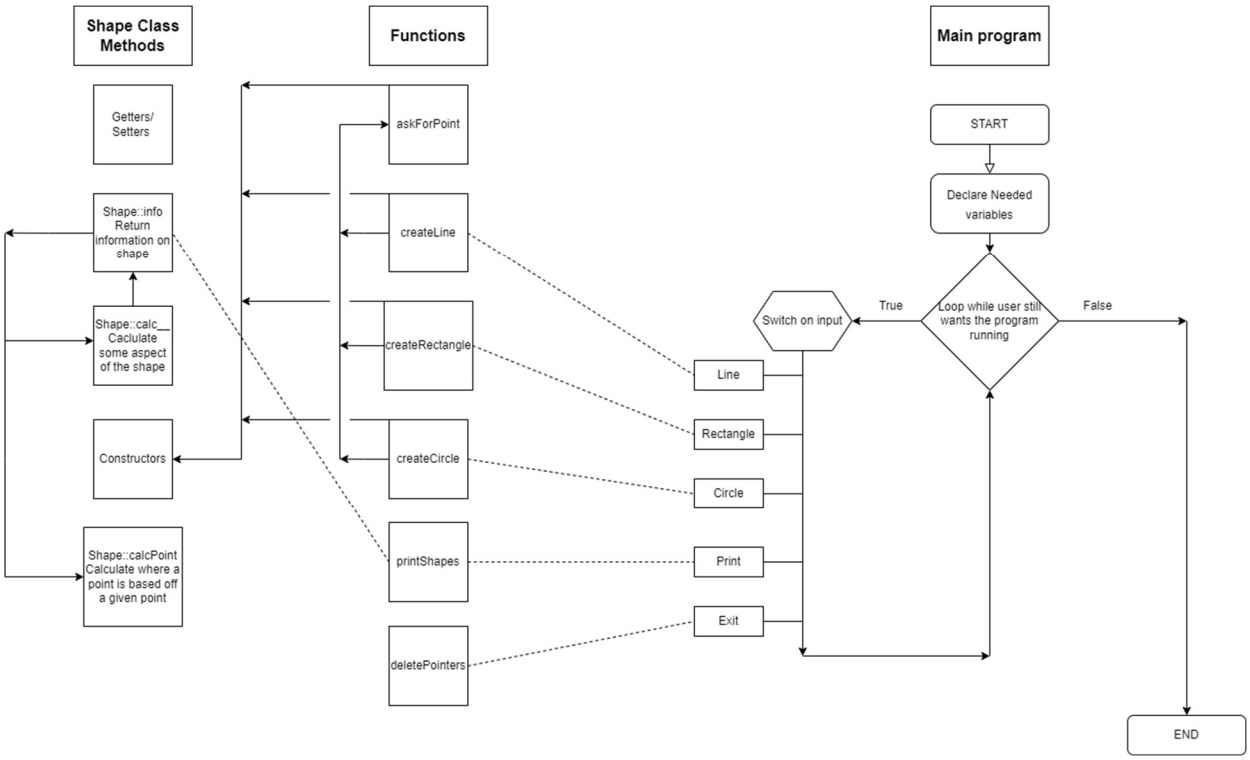
Rating	Definition of Risk Rating
Easy	The corrective action(s) can be completed quickly with minimal resources, and without causing disruption to the system or data
Moderate	Remediation efforts will likely cause a noticeable service disruption • A vendor patch or major configuration change may be required 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
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
Not Fixable	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 beta testing. This project was given to a user with minimal technology

4. Figures and Code

4.1.1 Process or Data flow of System



The user begins the program by selecting between 5 options:

- Create a line
- Create a rectangle
- Create a Circle
- Print the created shapes
- Exit

Depending on the selected option, a shape is crafted using the functions in the main running of the program. These functions use class methods from the associated shape class. When exiting, the program calls a function called “deletePointers” where all pointers created in the heap are deleted to prevent heap overflow.

5. Works Cited

Greenwell, Josiah. 2023. "Access Control Lecture." Estero, Febuary 2.

—. 2023. "Code Injection Lecture." Estero, March 30.

—. 2023. "Software Vulnerabilities and the OS Lecture." Estero, Apr 12.

Higgens, Seah. 2020. *Finding and Fixing C++ Vulnerabilities.* May 25. Accessed April 25, 2023. https://www.securecoding.com/blog/finding-and-fixing-c-vulnerabilities/.

Robert C. Seacord, Jason Rafail. 2005. "2005_017_101_52657.pdf." *cmu.edu.* Accessed April 25, 2023. https://resources.sei.cmu.edu/asset_files/Presentation/2005_017_101_52657.pdf.

University, Townson. 2022. *Townson University - C++ Buffer Overflow Injection.* Accessed Apr 25, 2023. https://cisserv1.towson.edu/~cyber4all/modules/nanomodules/Buffer_Overflow-CS2_C++.html.