**Logo, company name

Description automatically generated**

Pong Game

**Security Assessment Report**

May 1st 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)

1. Summary – The goal was to make my programming final, the game of pong a more secure game. This appeared to be challenging to me as the game was local and not really used to connect to others so that eliminated a lot of vulerbilites. There were some from the fact that it was uploaded to github.
   1. Assessment Scope

Windows 11

Google Chrome

Visual Studios

CLion

Chart, diagram

Description automatically generatedLacked multiopled operating systems

* 1. The largest Issued was the using namespace STD exploit
  2. The program was uploaded to a private github as a backup to the public one. Fixed the used of using namespace STD causing very large vulernabiies in the project. Fixed pointers, originally used them incorrectly. Made a physical backup on my hardrive incase the other fails.

1. Goals, Findings, and Recommendations
   1. The purpose of the assement was to expand our knowledge of security threats and how many we would have in an average project we did.
   2. Incorrect Pointers, vulnrebaties in namespace, used namespace std(common exploit), public github which left the source code vulernable, private github created to help combat this, but if my github account was hacked this would create another vulrebaitly, smart poitners used, physical backups of source code made in case compromside.
   3. Correctly use smart pointers as need, if using normal pointers I would implement them correcty. Correct a main public github and a private one. Also create physical copies on multiple drives incase any of them are to fail.

Methodology for the Security Control Assessment - Describe (over just list) tools and process used

* 1. Timeline

     Description automatically generated with medium confidence

Made sure that the project now is properly backed up and including all files so I was right in my assement. SFML is still causing major errors across different updates so it is essential to be on the same patch as the game was made on.

* 1. This section **will then have 3 parts** depending on your assessment and project. Some of the most likely elements would be:
     1. White box testing used
     2. sAnalysis of test results or Research into system vulnerabilities
        1. Using namespace std was the most common exploit used. This is very common as users typical used this to prevent themselves from using STD:: before certain statements. But usingnamespace STD causing a big vulernabilty
     3. Used a blind test of my mom like you showed in class using your son(she happens to love pong)

1. Figures and Code - Include any diagrams used for project here, tables, graphs, etc. where they don't fit in main sections (or break the reading flow).

Text

Description automatically generated

1. Works Cited - Cite your work (including code if taken from other user/site). **Must have 6 citations**(to include lectures, TryHackMe lessons, outside resources, code, etc.)

<https://tryhackme.com/room/introtooffensivesecurity>

https://developer.valvesoftware.com/wiki/Source\_Bug\_Reports

<https://tryhackme.com/room/learnowaspzap>

https://github.com/RequirementsEngineeringTemplate/website\_planning/tree/main/usecases

[**https://tryhackme.com/room/pentestingfundamentals**](https://tryhackme.com/room/pentestingfundamentals)

**http://agiledata.org/essays/tdd.html**