**Logo, company name

Description automatically generated**

**Security Assessment Report**

AAA

Version N.0

Angel Hernandez

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)

# Summary

The overall goal of this project was to analyze and indentify potential vulnerabilities or exploits in an old project. Most of the vulnerabilities I found were backdoors that gave the player the adavantage.

## Assessment Scope

The platforms I mostly used were visual studio community, youtube, and w3schools. The only OS that works for my project is Windows as this old project was entirely done using windows froms. I tested it on 2 different versions of windows 10, and also windows 7 and 11.

## Summary of Findings

Of the findings discovered during our assessment, 2 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 2.

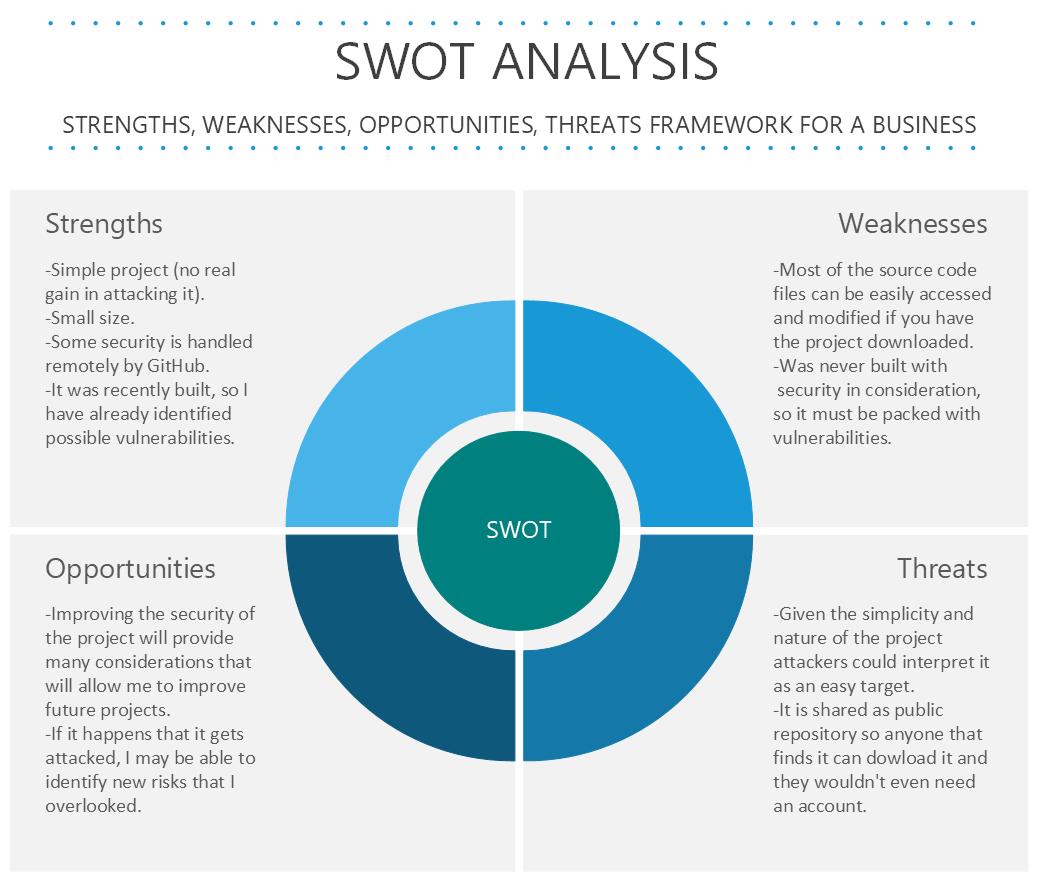


Figure 2. SWOT

The 2 high risks were 2 backdoors I never removed at the time because they were very hidden and even went unnoticed in some of the blackbox tests I conducted for this assesment. The 2 moderate risks were some rare or difficult to find unhandled exceptions that either crashed the program popped up an error window. The low risk was a small logic error that was not affecting the functionality of the program and was only indetified and fixed for if I wanted to to exapand the program in the future.

## Summary of Recommendations

Most of the changes involved organizing data, fixing a few exceptions, removing unnecessary code, functionality improvements and removing things I left there during the initial debugging.

## Assessment Goals

The purpose of this assessment was to do the following:

* Ensure that the game was in working as the orginal rules of the card game states.
* Determine if the game was securely maintained.
* Ensure that the player cannot break the system to gain advantage or win

## Detailed Findings

1) Removed a backdoor that allowed the player to display the dealer’s second card that is supposed to remain hidden until you press stay and the match is decided. This was by far the biggest security risk since the rules of the dealer only allows the dealer to reveal another card if and only if the sum of the first two cards is less than 17. This means that if the player knows the second card also knows if the dealer revelaed a third one or stayed because the sum of dealer’s first two cards was more than 17.

2) Removed a second backdoor that displayed the total number of cards dealt. This game (BlackJack) uses a standard 52-card deck. This resulted in the player gaining a slight edge since the program will always deal the same cards to both the dealer and player in the same order when the deck run out of cards. Only restarting the program will give you new cards.

3) Fixed an exception that caused an error when player pressed some cards. All the cards were buttons and some still had button press events linked to nothing or debugging code. All cards are supposed to be buttons only in name as their purpose was not to be pressed but to just hold an image of either a card or a cardback.

4) Fixed an exeption that caused a crash when both the dealer and player busted over the desired 21 value.

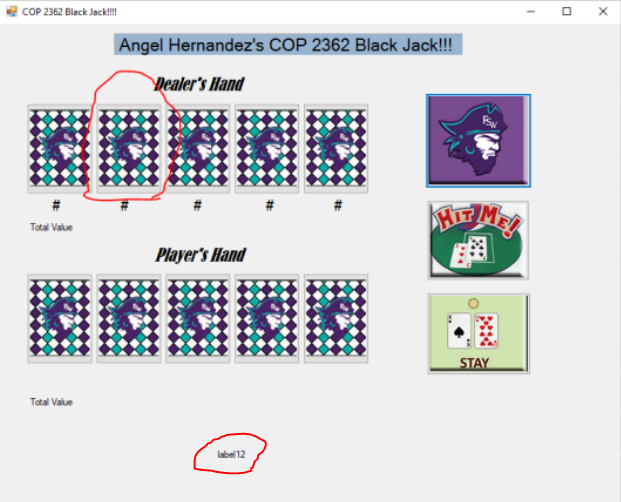
5) Fixed a small logic error that was not affecting the program or how the game played in any way but it was for a possible feature that could be added if I decide to keep working on this project.

6) Removed two uncesary variables that held information that was uncesary as it was part of a feature that was never fully implemented.

7) Noted that this application is terrible in the sense that is near imposible to port to other Operative systems as it was built using Windows Forms. This aspect was imposbile to fix because the entire thing was made using Windows Forms but it was indetified notheless.

8) Noted that there was no ruleset or tutorial of how the game works or what it does. There many variations of the game of black jack and this one uses the Vegas ruleset. Most people should be able to identify this but I did run through some of these issues during black box testing.

## Recommendations



# Methodology for the Security Control Assessment

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

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**

Two major types of tests were performed:

1. Black Box testing: this test involved making someone play the game for the first time without any previous knowledge of how the source code or how things were working in the background. This test originally felt redundant because for the original project I received a good score from my professor and everything worked just like he wanted. Notheless, I was proved wrong as it was during this test that I was able to find one backdoor and one of the testers found the other one among other things like a crash and different types errors. These went unnoticed at first because I never thought of doing things that were not intended to be done. Three persons from variying ages helped me performed this test.
2. White Box testing: this test involved playing with full or partial knowledge of the game being tested. This test helped me indentify different smaller issues like quality of life improvements, deleteig unecesary things and even a moderate one that involved a difficult to replicate result where both the dealer and the player busted over 21. This test was only performed by me and friend remotely using discord as finding other people that knew programming was no easy task.

**3.1.3 Tools**

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

1. Visual Studio: visual studio was the main tool used for testing. I have been using it for a long time and I’m very familiar with it. This helped me test and run code more efficiently and faster.
2. Discord: disocord was used to send code and files to some testers that helped me during Black Box and White Box tests.
3. Microsoft Word and Excel were also used to develop documents, tables, graphs and information that was vital to understand and record the data of all the tests.

# Figures and Code

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

### Other figure of code

# Works Cited

“Faculty/Staff Profiles | Florida SouthWestern State College.” *Www.fsw.edu*, www.fsw.edu/directory/user/rwebster. Accessed 1 May 2023.”) (Starting original code from Roger W. Webster – FSW Professor).

W3Schools. “W3Schools Online Web Tutorials.” *W3schools.com*, W3Schools, 2023, [www.w3schools.com/](http://www.w3schools.com/).

adegeo. “What Is Windows Forms - Windows Forms .NET.” *Learn.microsoft.com*, learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-7.0.

HowTo, Las Vegas. “Playing Blackjack in Vegas.” *Vegashowto.com*, [www.vegashowto.com/blackjack](http://www.vegashowto.com/blackjack).

Greenwell, J. “Educator and Engineer - Academic Resources.” *Www.jsgreenwell.com*, www.jsgreenwell.com/products-services. Accessed 2 May 2023.

GitHub. “GitHub.” *GitHub*, github.com/.