Integration Project
Security Assessment Report

Version N.0

May 1, 2023

Table of Contents

1. Summary 3
    1. Assessment Scope 3
    2. Summary of Findings 3
    3. Summary of Recommendations 5
2. Goals, Findings, and Recommendations 5
    1. Assessment Goals 5
    2. Detailed Findings 5
    3. Recommendations 6
3. Methodology for the Security Control Assessment 7
4. Figures and Code 10
        4.1.1 Process or Data flow of System (this one just describes the process for requesting), use-cases, security checklist, graphs, etc. ..... **Error! Bookmark not defined.**
        4.1.2 Other figure of code 11
5. Work Cited 11

1. Summary

The overall goal is to test the integrity of the Integration Project in terms of security and functionality.

1. Assessment Scope

For the sake of evaluating the constraint is limited to a computer with Windows 10 or 11. In addition the test will be conducted using Visual Studio Code and will be using Python. Research will be conducted using Google as our search engine. The system intended for testing is made through Python and visual studio code for text files to be readable.

2. Summary of Findings

Of the findings discovered during our assessment, 4 were considered High risks, 1 Moderate risks, 3 Minor, and 1 Informational risk. The SWOT used for planning the assessment are broken down as shown in Figure 1.

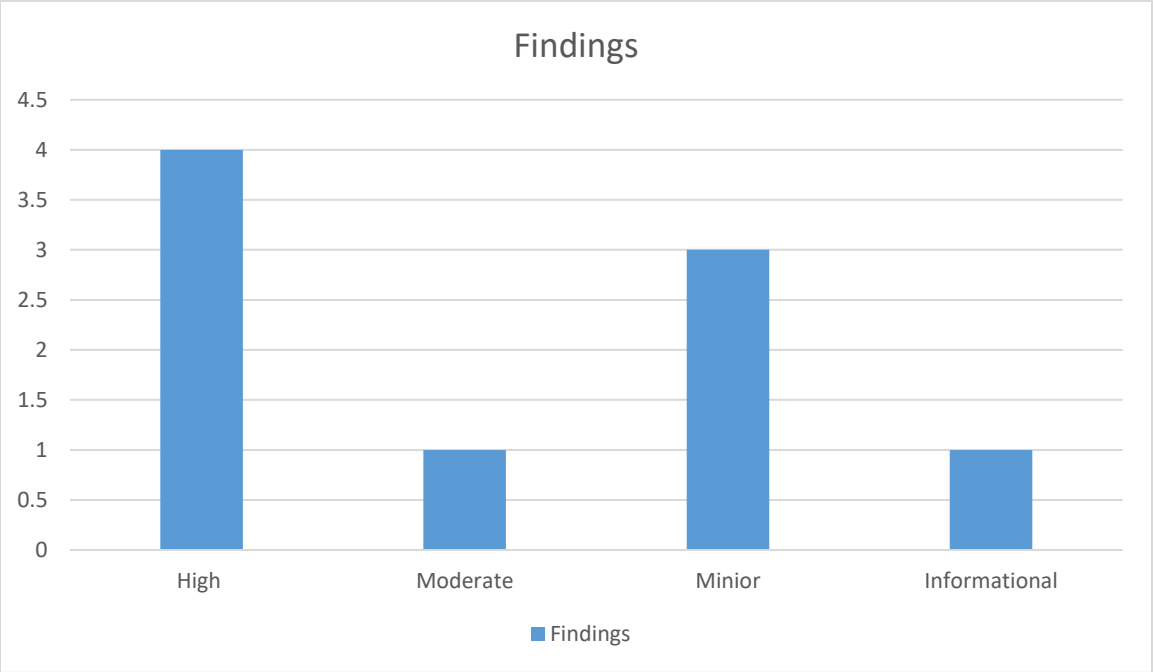


Figure 1. Findings by Risk Level

Major issue found is the integration project is outdated. The project itself was made in Python 3.8.5 which can result to the system failing to run. Another issue found user would have access to the text files that the project uses. This can result in deletion or temperament of the files, in addition failing to run due to no access or invalid context of the menu. The last issue found would be

inaccurate calculations. However, the majority of the minor tasks and informal task were already handled or will be planned to be enabled later on.

Figure 2. SWOT

Issues that will be addressed are the integration project is outdated. The project itself was made in Python 3.8.5 which can result to the system failing to run. Another issue that will be addressed is the user would have access to the text files that the project uses. This can result to deletion or temperament of the files, in addition failing to run due to no access or invalid context of the menu. Lastly, mishandled input by the user leads to inaccurate calculations. For the most part most of the issues have been managed and some can be implemented.

3. Summary of Recommendations

Changes that have been made were refining its exception handling and adding a .gitignore to hide my external files. For exception handling, there was a portion of the program that resulted in a value error as it could not handle chars such as a, x # etc. and should only use integers and the solution used was to input a try-except statement. Another would be the customer's calculations as there was no check to see if the customer paid the exact amount and therefore a check was implemented for it. Lastly including a .gitignore by creating it within the GitHub repository with a specific template. Some of the other issues(being it requires an IDE, code exploitation, and lack of backup) was mainly due to the project being stored in a repository in GitHub, and due to being the only user account that has access controls to the repository the account itself already has 2-Factor-Authentication and have a secure connection. Some changes that I have not made nor addressed was logging user input and files, doing encryption for the external files and creating a key for it, creating a template for the welcome function and re-writing the whole code to handle inputs well, therefore these will be handled in due time.
The current changes that the project still needs is to re-write all the legacy code. As mentioned, it was written in Python 3.8.5 in 2021 so it may not function the same way as it did before, and with the experience I currently have I could always re-write to be slightly better. In addition, the program has a lack of logging as it was a project made in Intro into Computer Science.

2. Goals, Findings, and Recommendations

1. Assessment Goals

The purpose of this assessment was to do the following:

- Determine if the application is secured.
- User input is handled.
- Matches with the requirements criteria given through each iteration.

2. 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.
Issues found:
-Text files are modifiable-Threat
-Text files can be deleted-Threat
-Incorrect Calculation-Weakness
-IDE required to run the program-Weakness

- Code is Exploitable-Opportunity

Recent Vulnerabilities:

-Input handling-Strength
-Legacy Code (Outdated.)-Threat
-Lack of backup
-Lack of Logging

Risks	Risk Ratings	SWOT Vulnerability	Definition of Risk Rating
Text files is modifiable	High Risk	Threat	The project uses text menu to read all the menus for readability and input. The menus are formatted in such a way to be displayed however the text file is written in plain text and are stored in the same folder as the program, and the program uses fixed values based off the menu if the text files were modified the result will lead to the program being functional however the calculations are inaccurate.
Text files deleted	High Risk	Threat	The project uses text menu to read all the menus for readability and input. The menus are formatted in such a way to be displayed however the text file is written in plain text and are stored in the same folder as the program, and the program uses fixed values based off the menu if the text files were modified the result will lead to the program failing.
Input Handling	Observational Risk	Strength	The project requires user input in order to calculate the total cost. Input handling is checked and can account for bad input.
Legacy Code	Low Risk	Threat	The project itself was made in Python 3.8.5 which can result to the system failing to run.
Code is exploitable	Informational risk	Opportunity	The financial calculation is determined by the fixed values within the project. If access control is given the values and operations can be modified resulting to inaccurate calculation or the program will fail to run.
IDE is required to run	Informational risk	Weakness	In order for the project to function, the project requires Python(System interpreter) to be installed, otherwise the program will not run
Inaccurate calculations	Informational risk	Weakness	The project calculations are done through fixed floating-point values, which can lead to inaccurate calculations that may overshot, or under shot the expected calculation
Lack of Logging	Moderate risk	Threat	The project requires user input in order for the program to run, it however does not account for invalid input
Lack of Backup	Informational risk	Weakness	The project is currently stored in a repository in GitHub. The original project was originally created on an older desktop and the data was lost during a full reset, currently it lacks a backup.

3. Recommendations

Risks	Fix Difficulty	Recommendation
Text files is modifiable	Moderate	Since the text files are considered external files, the files can be encrypted through Asymmetric encryption (Which may require more time to do so). In addition, the project is also saved on GitHub therefore adding a .gitignore is necessary.
Text files deleted	Moderate	Since the text files are considered external files, the files can be encrypted through Asymmetric encryption (May require more time to do so). In addition, the project is also saved on GitHub therefore adding a .gitignore is necessary as it will hide specific files per project.

Risks	Fix Difficulty	Recommendation
Input Handling	Easy	Input handling can be done through using Try-Except, or if, Elif, else statements. According to Python Try-except is preferred as it catches errors efficiently.
Legacy Code	Easy	Because the code was created in 2020 as an Integration Project, the code can be modified in order to perform the same operation, with small security risks.
Code is exploitable	Easy	To prevent the code to be exploited, some options is to save it in a more secure environment or repository, with limited access control. (Handled)
IDE is required to run	Easy	It is best recommended to run a Python IDE with a Linter. Recommendations would be visual studio, PyCharm etc.
Inaccurate calculations	No known fixes	The project calculations are done through fixed floating-point values. Though its built-in on purpose.
Lack of Logging	Easy	Python has a built-in library to log data, so a function can be created to log
Lack of Backup	Moderate	The project is currently stored in a repository in GitHub, so it can be easily accessible by the user with Two-Factor Authentication.

Listed version of Recommendations:

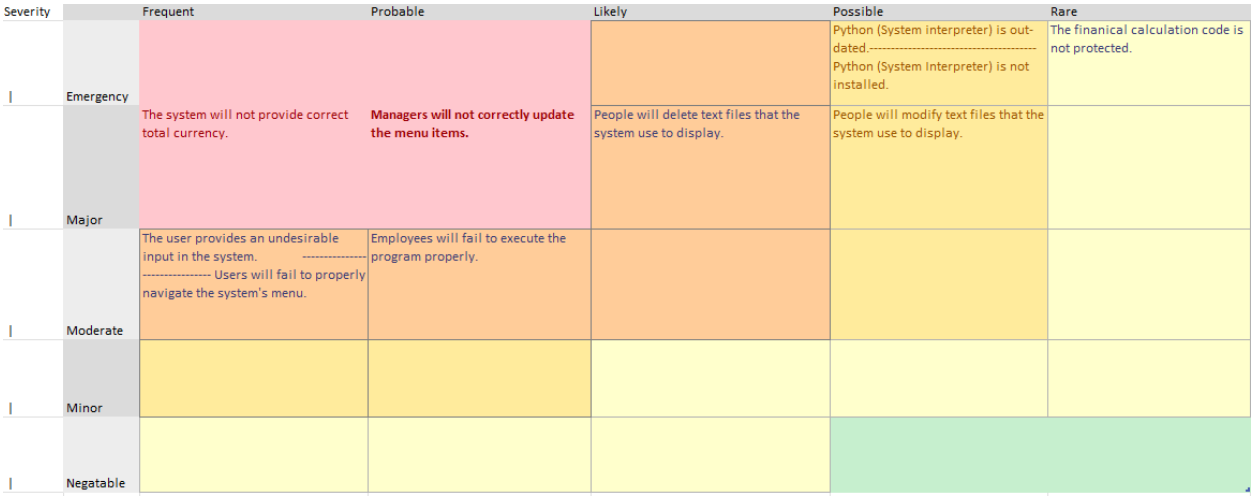
1) (Lack of Backup)-A recommendation for Lack of back up is to store the project in a secure cloud-system repository. This is also a benefactor to prevent code exploitation. When using an Online repository like GitHub, it restricts the amount of access that users have to a repository. GitHub can hide a repository that shouldn't be shown to the public and a solution can be creating a .gitignore file to specifically hide certain files included within the repository. In addition, GitHub tracks any commits made to the repository, and any case of traffic made on the repository.
2) (Legacy Code) Since the code is old, a recommendation would be to re-write the code from scratch. Old or legacy code can result to security loopholes or exploitable code that can lead to security vulnerabilities. Re-writing the code can improve input handling, revealing and preventing loopholes that users may use, an easier way to log input from the user, and lastly, a more efficient way to run the program.
3) (Input Handling)-Refer to 2). To add on, input handling can catch errors through try- except checks or using if/else statements for unnecessary values.
4) (Text Files be modified and deleted) – To prevent text file from being modified it's best to encrypt the files using Asymmetric Encryption which prevents users from modifying the text files and can only be opened with a password. Refer to 1) for the second reason.

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".

Security Assessment – Integration Project



Severity		Frequent	Probable	Likely	Possible	Rare
I	Emergency				Python (System interpreter) is out-dated.---------- Python (System Interpreter) is not installed.	The finanical calculation code is not protected.
I	Major	The system will not provide correct total currency.	**Managers will not correctly update the menu items.**	People will delete text files that the system use to display.	People will modify text files that the system use to display.	
I	Moderate	The user provides an undesirable input in the system. ---------- Users will fail to properly navigate the system's menu.	Employees will fail to execute the program properly.			
I	Minor					
I	Negatable					

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

Rating	Definition of Risk Rating
	• 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

White-box testing:

The purpose of the test to check and verify if the legacy code still functions as it intended to in Python 3.9 and that user input is handled properly. Throughout the test, the project is still functional in Python version 3.9 however, it fails to handle two exceptions. As previously described, the program that handles the main dish quantity threw a value error if a char or other ASCII symbols were inputted. The other exception was that it didn't account if the customer paid the flat amount. In addition, the user can type in a command in the welcome function as it lacks a check to verify if the user is typing their name. The program still has an issue with dealing with integers less than zero after going through exception checks.

3.1.3 Tools

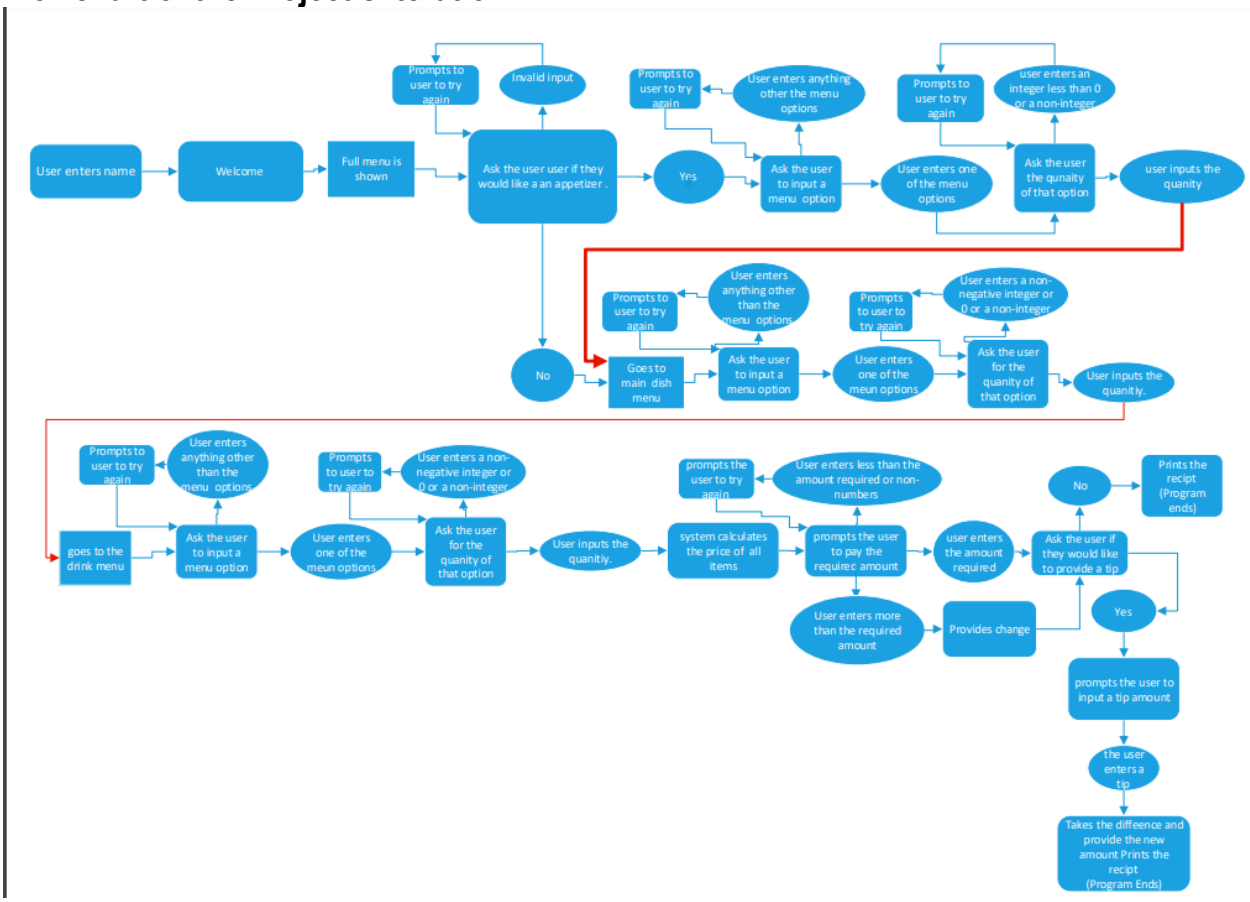
This was completed using:

- Visual Studio Code (Python Version 3.9)
    - IntelliSense (Pylance)

4. 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:

4.1.1 Flowchart of the Project's Iteration



The flow chart explains the process of the code of when user enters their name to the printing of the receipt. The flow chart also shows if the user provides any alternative or invalid inputs.

4.1.2 Solution for Exception Handling (Value Error)

```
while True:
    try:
        main_dish_qt = int(
            input("How many servings would you like?: "))
        break
    except ValueError:
        print("Improper Input. Please provide a whole number input.")
if main_dish_qt < 0:
    print("Improper input. Please input a whole number.")
    while True:
        try:
            main_dish_qt = int(input("How many servings "
                                     "would you like?: "))
            break
        except ValueError:
            print("Improper input. Please input "
                  "a positive whole number.")
    while True:
        try:
            main_dish_qt = int(input("How many servings "
                                     "would you like?: "))
            break
        except ValueError:
            print("It must be an integer that is greater than 1.")
```

This code snippet was the solution based off the exception issue caused by a value error.

4.1.3 Exception Handling(Unaccounted check)

5. Work Cited

Greenwell, Josiah. “CIA Triad” March 14, 2023

badges, JoyJoy 2111. “Git - Create a .Gitignore File after Creating the Repository and Publish .Idea File? - Stack Overflow.” Stack Overflow, https://stackoverflow.com/questions/63972265/create-a-gitignore-file-after-creating-the-repository-and-publish-idea-file.

"Built-in Exceptions — Python 3.11.3 Documentation." Python Documentation, https://docs.python.org/3/library/exceptions.html.

github. "GitHub - Github/Gitignore: A Collection of Useful .Gitignore Templates." GitHub, https://github.com/github/gitignore. Accessed 1 May 2023.

Prokopets, Marie. "How To Add A File To Gitignore On Github." Nira, 11 May 2022, https://nira.com/how-to-add-a-file-to-gitignore-on-github/.

"Python Language Basics: Understanding Exception Handling | Infosec Resources." Infosec Resources, https://resources.infosecinstitute.com/topic/python-language-basics-understanding-exception-handling/.

"When to Use Try/Catch Instead of If/Else - PythonForBeginners.Com." PythonForBeginners.Com, https://www.facebook.com/pythonbeginners, 24 Mar. 2021, https://www.pythonforbeginners.com/control-flow-2/when-to-use-try-catch-instead-of-if-else.