Andrew Atkins

12/6/2024

Andrew Atkins Final Project Write Up

This project was in my past programming one class. I was tasked with writing an encryption program for an IT company. The program should give the option to either encrypt or decrypt a file of passwords. The encryption option should take each password in the file, encrypt it, and output the encrypted contents to a new file. The decryption process should reverse a previously encrypted file and output the original message to a new file. The goal of this program was to make sure that the passwords could be decrypted so that they could not be stolen by malicious users. Unfortunately, at the time when I made the project, I was not familiar with the concepts that we have learned in this course, and due to this my original code did not have any input validation, error handling, tests to run the code, or verification of file paths. During this final project, my goal was to both make the project more efficient with tests and to make sure that malicious users could not input malicious files or code to find the users passwords. To do this, I made tests so that we could better understand and make my code more efficient, changed my input validation process to ensure file paths are both existing and not malicious, added error handling so that when errors present themselves, we can handle them correctly and efficiently, finally I ensured that the users input, and files were proper to numbers and not malicious code. All of these steps are depicted below and gone through more in depth.

A black background with yellow and orange letters

Description automatically generated

To begin, I changed parts of the code to add for input validation for file paths to ensure that users give us a both valid and existing file path, as well as making sure it operates securely and reliably. This input validation will prevent issues like not found errors, unauthorized access, and directory traversal attacks. By checking the user input and adding the function “validate\_file()’ we can prevent malicious data that could compromise functionality or lead to vulnerabilities by checking if the file exists and whether the file path is correct.

A black rectangle with white text

Description automatically generated

A black rectangle with white text

Description automatically generated

Secondly, I added error handling to manage unexpected scenarios with missing files, invalid data, or permission issues. By wrapping files in try- except blocks it makes sure the program doesn’t crash and provides the reason for the error so that the user can fix the problem. By wrapping files in try- except blocks we can catch the FileNotFoundError and the PermissionError.

A screenshot of a computer

Description automatically generated

Next, I made sure that the entered values are proper numeric values through input validation. This maintains the data’s integrity, prevents runtime errors, and gives us a safety net for corrupted data. By using the Value Error exception that is already built into python, we can throw an error when given improper numeric values to further prevent against invalid entries.

A screenshot of a computer program

Description automatically generated

Finally, I installed tests to ensure the code works as intended by verifying its functionality after updates and changes. These tests help me catch bugs and validate edge cases in my code so that I can improve on its readability. It allows for multiple people to edit the code confidently because tests safeguard against errors. I validated all parts of the code from validated file handling with validate\_file, and the encryption and decryption process using pythons built in unittest framework that covers most errors.

The changes in the code outlined above improved on the overall security of my past project. By validating file paths from the user, adding error handling, validating numeric values entered by the user, and implementing tests, the project has improved significantly from when I first created it. I believe the changes I have made to the code reflect everything we have worked on over this semester.