Introduction:

This test plan is a Master-level plan for the Windows OS and Application Vulnerability Scanner. It outlines what will be required for the testing process, as well as what kind of testing will be done. It also details the items associated with the testing process, like relevant documentation and testing concerns.

- Goals:
    - Clearly define the testing process and environment.
    - Provide easy-to-understand test criteria
    - Ensure user data safety and prevent harm

- Constraints:
    - Time – I must divide my time between my other studies and responsibilities and testing my program.

References:

- The following documents are relevant to the program and its development and testing, and can be found at https://github.com/DylanAKelly/CSU-Senior-Project/tree/master:
    - SeniorProjectDraftProposal-DylanKelly.md
    - SeniorProjectRequirementsDocument-DylanKelly.md

Test Items:

- Python (v3.13)
- Visual Studio Code (v1.94)
- Windows OS and Top 20 App Vulnerability Scanner

Features to be Tested:

- User OS (Windows) Vulnerability Scan
    - Requirements Document ID Number VS-REQ-11

- User Top 20 Application Vulnerability Scan
    - Requirements Document ID Number VS-REQ-12

- User OS (Windows) Update Check
    - Requirements Document ID Number VS-REQ-13

- User Top 20 Application Update Check
    - Requirements Document ID Number VS-REQ-14

- Accept, Store, and Retrieve User Email
    - Requirements Document ID Number VS-REQ-15

- Send Email Alert When Vulnerabilities are Found or an Update is Available
    - Requirements Document ID Number VS-REQ-16

Features Not to Be Tested:

- Visually Appealing
    - A traditional test case cannot be created for this feature. Instead, an external survey will be used to objectively measure this subjective requirement.

- Responsive to Commands
    - A traditional test case cannot be created for this feature. Instead, an external survey will be used to objectively measure this subjective requirement.

- Easy to Set Up, Navigate, and Use
    - A traditional test case cannot be created for this feature. Instead, an external survey will be used to objectively measure this subjective requirement.

- Complies with Modern Computer Science Ethical Standards
    - A traditional test case cannot be created for this feature. Instead, I will complete an ethics self-assessment document and review the findings of that document with an individual who has more computer science ethics knowledge than I do.

For more information, see the "User Acceptance Test Strategy" and "Manual Testing" sections of my Test Plan markdown file, located in the documents folder of my GitHub Repository.

The detailed Test Cases can be found in the project's GitHub Repository as well.

Approach:

- The overall approach to testing will see:
    - Testers review sections of the program on screen during the development process
    - Periodical testing checkpoints upon completion of a project deliverable
    - Tests conducted on multiple project parts in tandem

- Testing Level:
    - Master Test Plan

- Testing Types:
    - Manual Testing
    - User Acceptance Testing

- Testing Methods:
    - Dry-Run Testing
    - WhiteBox Testing

Item Pass/Fail Criteria:

- In order to pass a test case:
    - The output provided by the program must match the description and security risks of a given Windows OS or application version, which can be found on https://www.cvedetails.com OR
    - An email containing the correct, relevant information on their current Windows OS and application versions, as well as the security risks present in those versions is sent to a user

- Any other outcome will be seen as a failure.

Suspension Criteria and Resumption Requirements:

- All testing will be suspended immediately if:
    - Sensitive user data is found to be insecure OR
    - The program is unable to be rendered due to technical errors.

- The testing activities which must be redone when testing is resumed are:
    - All tests completed prior to the cause of suspension AND
    - The test that caused the suspension AND
    - Any remaining tests that have not yet been completed AND
    - All program security and integrity tests.

Test Deliverables:

- The deliverables for my testing process are as follows:
    - Test Plan (this document)
    - Test Cases
    - Test Reports
    - User Acceptance Survey
    - Ethics Self-assessment Document

Test Environment:

- Hardware:
    - AMD Ryzen 5 5600G with Radeon Graphics 3.90 GHz, 16.0 GB RAM, 465 GB Storage

- Software:
    - Windows 10 Pro, Visual Studio Code, GitHub

- Network:
    - Local household network

- Related Tools:
    - Python Modules (Unittest and/or Doctest)

Estimate:

- Cost:
  - There is no expected cost for the testing process

- Time/Effort:
  - The testing process is estimated to require three to seven hours of activity per week. The estimated effort for the process is estimated to be significant.

Schedule:

- 8/26/24 to 10/7/24:
  - Create master test plan

- 10/8/24 to 10/20/24:
  - Revise master test plan as needed
  - Begin test case construction

- 10/21/24 to 12/15/24:
  - Begin testing
  - Continue test case development
  - Log test results

Staffing and Training Needs:

- Staffing needs:
  - The only requirement for testing staff is to understand the language the program is written it.
- Training needs:
  - There are no training requirements. However, while testing, it is important to make sure the user's security is never compromised.
  - Additional testing tips and strategies will be implemented as I learn of them through my work in other classes.

Responsibilities:

- The lead project developer will be responsible for all testing.

Risks:

- The risks that have been identified are as follows:
    - User information used by the program (i.e. email, installed applications) could be exposed.
    - User information not used by the program could be unintentionally accessed and exposed (user account, sensitive personal or financial information)

- The mitigation and contingency plan is to:
    - Immediately shut down testing in the event a risk is identified.
    - Determine the cause of the risk.
    - Resolve the risk.
    - Document the process from beginning to end.

Assumptions and Dependencies:

- Assumptions:
    - The tester will have uninterrupted (physical and digital) access to the testing environment at all times.
    - The hardware and software used in the testing process will be reliable.

- Dependencies:
    - External: Internet Service Provider continues providing service
    - External: Power Company continues supplying power
    - Internal: Code modules remain functional
    - Internal: Code libraries remain functional

Approvals:

- My Senior Project Advisor Dr. Sean T. Hayes must approve my test plan.