Operating System & Top 20 Apps Vulnerability Scanner

Dylan Kelly
B.A. Applied Computing
(Cybersecurity Concentration)
Advisor: Dr. Sean Hayes
April 25, 2025

Statement of Purpose

- Provide users with a simple, easy to use program that can assist in increasing system security and threat awareness in everyday computer users.
- Alert & remind everyday users of vulnerabilities in their Operating System & installed applications, forcing them to think about their digital safety and become more aware of security best practices.

Problem Statement

- People often do not update their Operating System or installed applications as soon as an update is available.
- This can be due to a variety of reasons, including:
  - Forgetting to
  - Being too busy to check
  - Lack understanding of how vulnerabilities can affect them
- Failure to update OS or applications to the current version significantly increases the likelihood that a threat can exploit a vulnerability, which puts the user at risk.

Research and Background

- **Background:**
  - I wanted my project to relate to my field of study.
  - I determined that a vulnerability scanner was a good intersection of cybersecurity and practical coding.
- **Research:**
  - Re-familiarize myself with Python and learn more about the language and applicable libraries.
  - The program Grabber (a Python Vulnerability scanner) was my inspiration for this project, and was used as a reference.
  - Had to learn how to web scrape using BeautifulSoup, how to send emails from my program, and how to create a functional GUI using tkinter.

Project Language(s), Software, and Hardware

- **Language:** Python
  - Familiarity with HTML, though no HTML code was written
- **Libraries:**
  - BeautifulSoup - Web scraping
  - Multipurpose Internet Mail Extension (MIME) & smtplib - Sending the email
  - Tkinter - Application Graphical User Interface
- **Software:** Visual Studio Code
- **Hardware:** Windows 10 Pro PC and Laptop

Project Requirements

- **Vulnerability Scanner**
  - Scan's the user's Windows computer and determines the Operating System version and version of any installed top 20 applications.
  - Pulls information from CVE Details based on the user's OS and installed top 20 applications.
- **Email**
  - Email the scan results to the user once the scan is complete.
- **Other**
  - Program is acceptable in terms of look, feel, and performance to users.
  - Program does not violate industry ethical standards and guidelines.

Project Implementation

- **Scanning -** Uses platform and subprocess libraries to retrieve Operating System and application versions, respectively.
- **Web scraping -** Uses BeautfifulSoup to scrape vulnerability information from the CVE Details HTML pages for the OS and applications.
- **Email -** Uses Multipurpose Internet Mail Extension (MIME) and smtplib libraries to send the user an email containing their scan results.
- **Graphical User Interface -** Uses tkinter and threading libraries to create and update the GUI.

Test Plan and Results

- **Manual Tests**
  - Opted to perform manual testing over automated testing due to the nature and scope of the project, as well as time efficiency concerns.
  - Total of fourteen manual test cases, with test fourteen being an ethical evaluation of my program.
- **User Acceptance Tests**
  - Total of eight user acceptance test cases.
  - Testing consisted of a user survey, where test cases were represented by questions.
  - Each test case had to receive an average of four stars (out of five) in order to be considered as passing, and five users performed the evaluation.
  - Four of the five users could be considered as less technologically-inclined, while the fifth user was proficient with technology and computers.

Challenges Overcome

- Refamiliarizing myself with Python, specifically some of the language's more unique features.
- Researching and learning how to use BeautifulSoup to web scrape.
- Figuring out how to use the MIME and smtplib libraries to send the user an email.
- Learning how to use tkinter and create a GUI in Python.

Future Enhancements

- **Implementing a database element:** Would allow the user's email, vulnerability information, and the program's email to be stored in a more secure and appropriate way.
- **Use PowerShell more effectively:** Increase the efficiency and speed at which the program operates.
- **Split the functionality of the program:** Further improve program efficiency and provide the user with some degree of agency regarding how the scan is conducted.
- **Alter how the program retrieves data:** Help ensure that the data retrieved from web scraping is current information and would remove the need for manual changes.
- **Have program run on computer startup:** Guarantee that the scan is run more frequently, reducing the risk that the user is unaware of potential vulnerabilities in their system and apps.

Conclusion

- The Windows OS and Top 20 Applications Vulnerability Scanner helps users, particularly those who are not as familiar with technology and security best practices, be more aware of potential security risks on their Windows systems.
- The use of an easy-to-understand language, paired with an easily understandable and modifiable design allows for changes and improvements to be made relatively easily.
- My project showcases my ability to design and create a practical solution to a serious issue faced by many people, while still leaving the door open for meaningful changes and enhancements to be made.