VigilBoard

William Mahoney, David Abbot, Isaac Bamidele, Chance Currie

What is VigilBoard?

- VigilBoard is an application that provides a quick and easy way to run port scanning and vulnerability scanning on a target system with a reachable IP address or website URL.
- We want to give security professionals and IT professionals access to a simple-to-use, easy-to-understand tool without requiring knowledge about using a command line interface.
- Authorized individuals can utilize VigilBoard to provide insight to their system with an efficient and easy to read output.

Objective

- Systems will always have vulnerabilities of some kind, whether they are known or not, and it's up to those protecting those systems to be able to know what vulnerabilities are present and the best course of action to take when dealing with a vulnerability.
- There are existing tools out there that you can use for these purposes, but not everyone is able to easily understand how to use these certain tools, understand the information that is being presented to them after the tool is finished running, or fully understand the kinds of concepts that a IT or security professional should know when using these tools.
- Our tool, VigilBoard, aims to solve this problem by getting right to the point by taking in an IPv4 address or website URL with a very user-friendly GUI, utilizing port and vulnerability scanning methods behind the scenes to detect any potential vulnerabilities, then compile a nice, organized report of vulnerabilities found, as well as give the best recommendations in order to rectify the found vulnerabilities.

Design

- Our design is split up into two parts: an application and a website

- The application is the main portion of this project.
  - As mentioned previously, it will utilize port and vulnerability scanning methods to seek out potential vulnerabilities on a system and generate a report in a clean and easy-to-understand format.

- The website is a “test case”
  - The website is currently being used to as a test site for the tool to analyze
  - We also want the website to be a place where you can easily view the reports that are generated after the tool has finished running, but this will be a secondary objective if we have time after completing the application.

Implementation

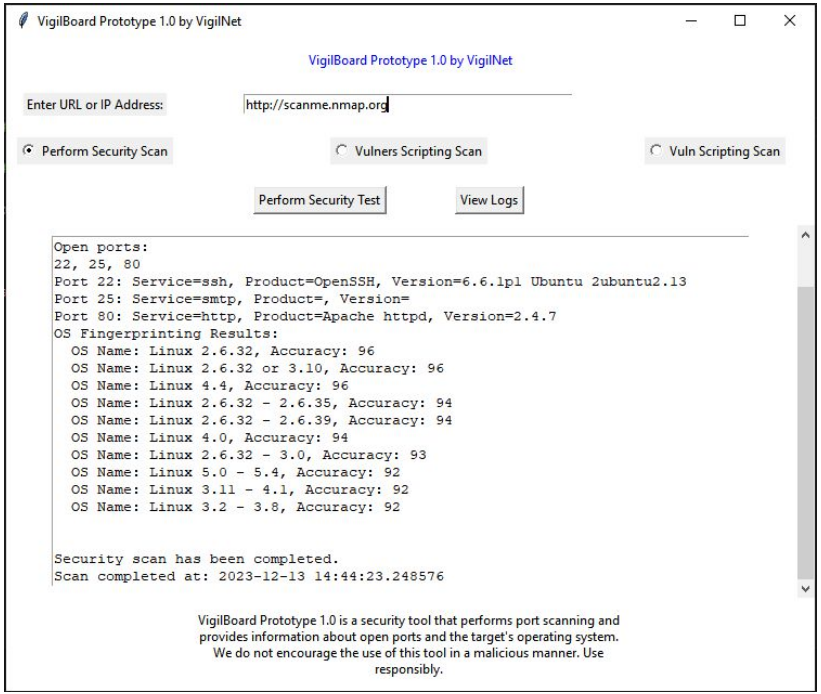
- The application implementation:
  - Coded in Python for easy integration of APIs and programming terminal commands into the application
  - Implementing nmap [1] through terminal commands for port scanning
  - Implementing vulners [2] through terminal commands for vulnerability scanning
  - Implementing tkinter (a Python GUI API) [3] for a user-friendly GUI when working with the application
- The website implementation:
  - Coded in typescript
  - Implemented basic authentication for user on the client-side.
  - Implementing user account storage.

Results

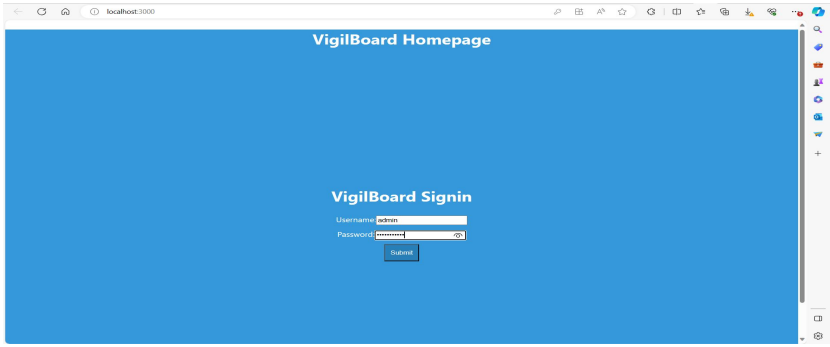
- We are happy to say that we are on track with our project.
- For our application:
  - We have port scanning and vulnerability scanning working on any given IP or website URL.
  - We are able to generate a log file and store it in the directory where the application is being ran when a scan is successfully completed, and we hope to translate this over into a bigger and better report generation feature for vulnerabilities in the next semester.
  - We have a GUI that works as intended and we hope to make it look more presentable in the next semester.
- For our website:
  - We have a basic website running as intended.
  - We have user authentication running as intended.

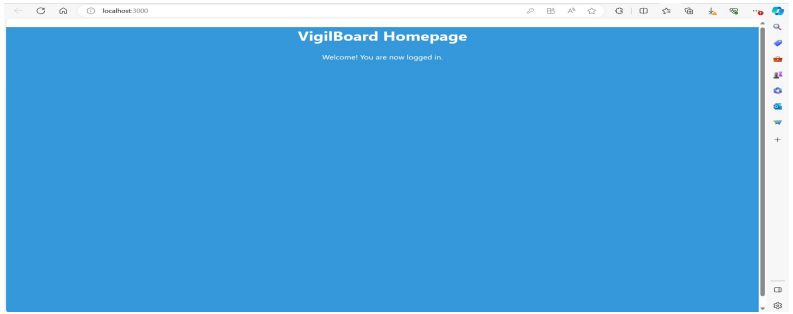
Results - Screenshots

Successful nmap port scan

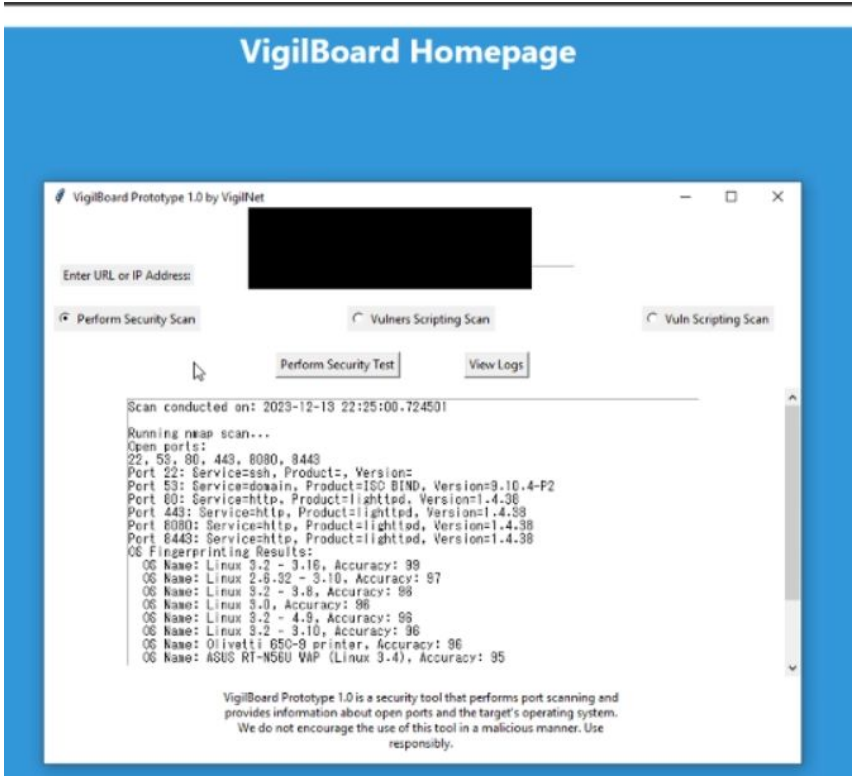


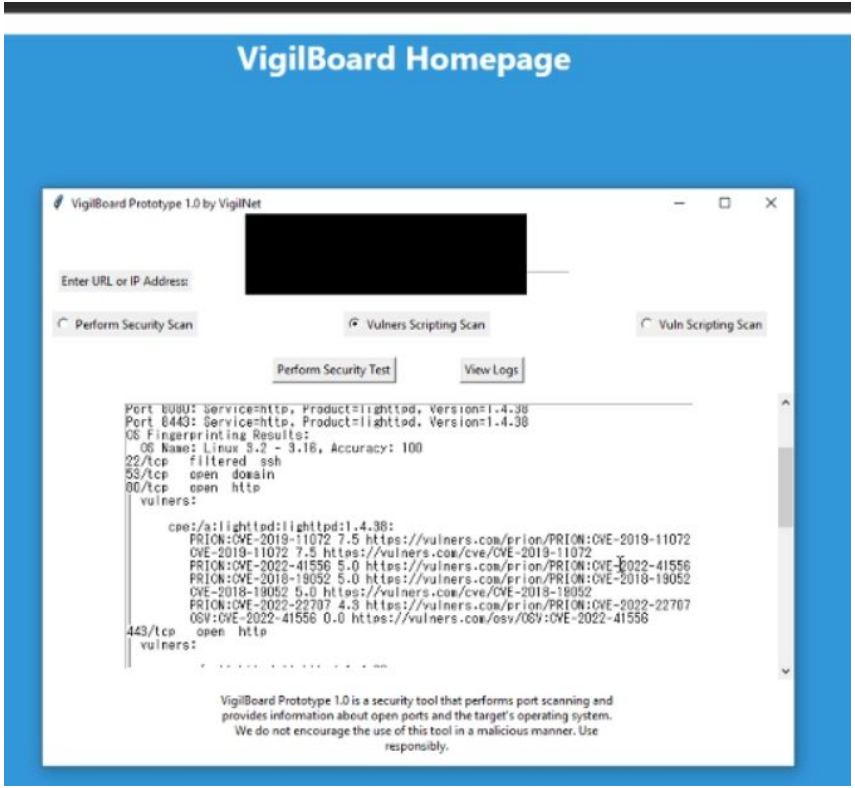
Website Home Page





Results - VigilBoard Homepage Website Scan w/ Application





Unit and Performance Test

scanlog2023-12-13 183945.txt - Notepad

File Edit Format View Help

hi

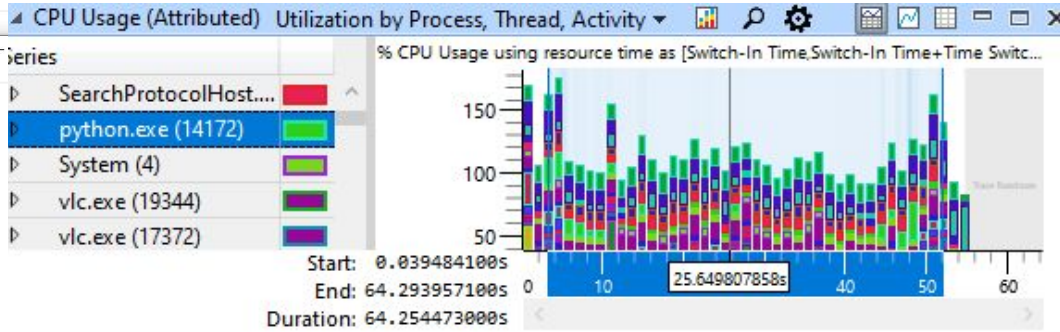
test_vigilboardmain.py - Notepad

File Edit Format View Help

```
from vigilboardmain import *

def test_try_to_write_to_file():
        save_to_file("hi")
```

Ln 4, Col 20 100% Windows (CRLF)

```
Successfully installed python-nmap-0.7.1
(base) PS C:\Users\Dave\documents\capstone\VigilBoard> pytest
============================= test session starts =============================
platform win32 -- Python 3.9.13, pytest-7.1.2, pluggy-1.0.0
rootdir: C:\Users\Dave\documents\capstone\VigilBoard
plugins: anyio-3.5.0
collected 1 item

test_vigilboardmain.py .

============================== 1 passed in 54.67s ==============================
(base) PS C:\Users\Dave\documents\capstone\VigilBoard>
```



Windows Performance Analyzer [4]

Results - Demo Video

Schedule - Application

Task Name	Report Generation Feature	General Application Building and Testing 2	More features, bug fixes, finalizing application development	Finalized GUI	General Application Building and Testing 3	Application Fully Completed
Expected Completion Date	02-01-2024	02-15-2024	04-04-2024	04-18-2024	04-18-2024	04-25-2024

Schedule - Website

Task Name	Fully implemented backend	Finished Website
Expected Completion Date	04-04-2024	04-18-2024

Conclusion

- We hope that VigilBoard will help people keep their systems safe and secure and make them aware of potential vulnerabilities, even if the people using the tool do not have much technical or security knowledge.
- The project and the process involved with it has been a great experience so far, and being able to do something like this is eye-opening and very educational in terms of learning the process of a secure development lifecycle and going through the necessary steps to ensure that what we are building can be considered a secure application
- We have a lot to work on for the website to have any functionality with the logs, and we still need to test databases, add more layers of scanning and testing.

Acknowledgements

- We would like to thank Dr. Shrestha for overseeing this project, as well as giving us excellent guidance and suggestions to improve our project. We will be counting on his continued support in the upcoming semester.

References

[1] Nmap, “Nmap,” *Nmap.org*, 2017. https://nmap.org/

[2] “Vulners - Vulnerability Data Base,” *vulners.com*. https://vulners.com/

[3] Python Software Foundation, “tkinter — Python interface to Tcl/Tk — Python 3.7.2 documentation,” *python.org*, 2019. https://docs.python.org/3/library/tkinter.html

[4] Windows Performance Analyzer, “Windows Performance Analyzer,” learn.microsoft.com, 2020 https://learn.microsoft.com/en-us/windows-hardware/test/wpt/windows-performance-analyzer

Any Questions?