National College of Ireland

Configuration Manual

Integrating Open-Source Vulnerability Scanning Tools Reports with Open AI API for Automated Report Generation

MSc Research Project
Cyber Security

Chinmay Dabholkar
Student ID: x21130680@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Chinmay Dabholkar
Student ID:	x21130680@student.ncirl.ie
Programme:	Cyber Security
Year:	2022
Module:	MSc Research Project
Supervisor:	Niall Heffernan
Submission Due Date:	14/08/2023
Project Title:	Configuration Manual
Word Count:	346
Page Count:	4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	17th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Integrating Open-Source Vulnerability Scanning Tools Reports with Open AI API for Automated Report Generation

Chinmay Dabholkar
x21130680@student.ncirl.ie

1 Introduction

Welcome to the Configuration Manual for the Vulnerability Report Summarizer Web Application. This guide empowers you to efficiently set up and utilize this tool, simplifying the process of analysing security vulnerabilities. By combining Nessus and OpenVAS reports and leveraging AI, this application generates concise summaries. Whether you're a developer or new to web apps, our clear instructions and code snippets will assist you. You'll learn Flask setup, HTML template design, and OpenAI API integration, enabling automated report generation. By the end, you'll confidently deploy and operate the Vulnerability Report Summarizer, streamlining your vulnerability assessment process.

2 Configurations

2.1 Hardware

- Operating System: Kali Linux, Windows
- Processor: Ryzen 7
- Architecture: 64bits
- Storage: 1TB HDD, 512GB SDD
- Memory: 16GB

2.2 Software

- Visual Studio Code
- Python
- Flask
- OpenAI API
- Pandas and NumPy

- Bootstrap(optional)
- BeautifulSoup
- Web Server
- Openvas
- Nessus

3 Implementation

Step 1 - Scan network by using Nessus and OpenVAS open source vulnerability scanning tools. In my case I have scaned Metaexploitable server. (1) (2)

Step 2 - Save the generated report in CSV format.

Step 3 - import and install these modules.

```
 import os
 from flask import Flask, request, render_template
 import API
 from pathlib import Path
```

```
import pandas as pd
import numpy as np
```

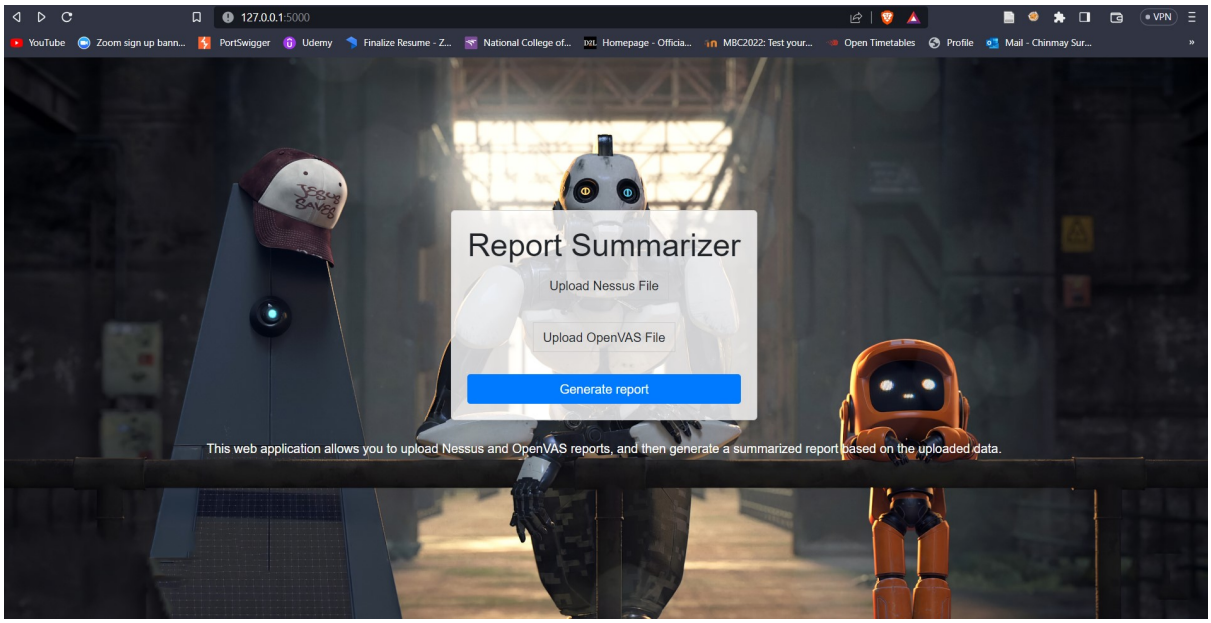
```
import openai
from bs4 import BeautifulSoup
import json
import combinecode
```

(3) (4) (5) (6) (7)

Step 4 - Run uploadFile.py

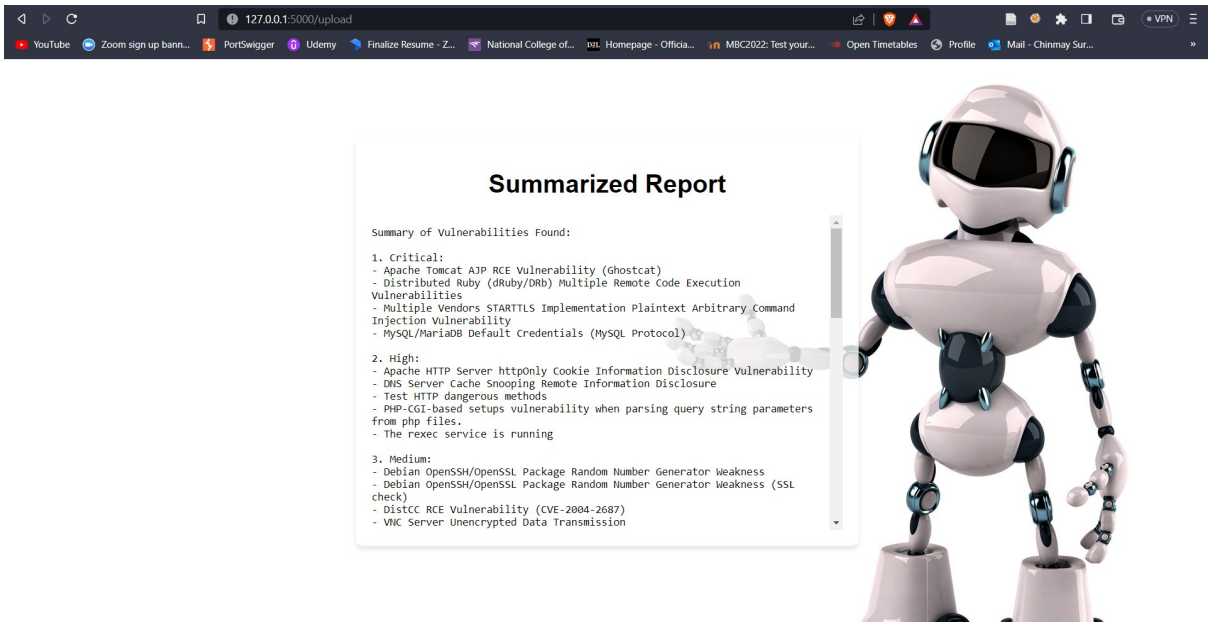
```
PS D:\NCI\sem 3\research method\Project\SumRep>  & 'C:\Users\csdab\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\csdab\.vscode\extensions\ms-python.python-2023.14.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '18259' '--' '-m' 'flask' 'run' '--no-debugger' '--no-reload'
 * Serving Flask app 'upload_File.py'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Step 5 - On browser start the web server with this link http://127.0.0.1:5000



Step 6 - upload both the file in respective fields and then click on Generate report button the pop up will asire then click on ok.

Step 7 - You will see AI generated report.



References

[1] OpenVAS. [Online]. Available: https://github.com/greenbone/

[2] nessus, "Getting started with nessus on kali linux," May 2019. [Online]. Available: https://www.tenable.com/blog/getting-started-with-nessus-on-kali-linux

[3] OpenAI. [Online]. Available: https://platform.openai.com/docs/api-reference/chat

[4] Flask, "Flask installation." [Online]. Available: https://flask.palletsprojects.com/en/2.3.x/installation/#virtual-environments

[5] Matthijs999900, "Os-sys." [Online]. Available: https://pypi.org/project/os-sys/

[6] python, "The python tutorial." [Online]. Available: https://docs.python.org/3/tutorial/index.html

[7] V. Studio, "Ide and code editor for software developers and teams," Aug 2023. [Online]. Available: https://visualstudio.microsoft.com/