Configuration Manual

MSc Industrial Internship
Cyber Security

Yogesh Anandhakumar
Student ID: x23167998

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Yogesh Anandhakumar		
Student ID:	x23167998		
Programme:	MSc Cyber Security	**Year:**	2023-2024
Module:	MSc Industrial Internship		
Lecturer:	Kamil Mahajan		
Submission Due Date:	02/09/2024		
Project Title:	Hybrid Detection of Cross-Site Scripting (XSS) Vulnerability in Web Applications		
Word Count:	374	**Page Count:**	05

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: Yogesh Anandhakumar

Date: 02/09/2024

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):	

Configuration Manual

Yogesh Anandhakumar
Student ID: x23167998

1 Introduction

This document is configuration manual of the tool, it consists of necessary details of the proposed tool. It has all the steps with instructions to run the tool. This document consists of all the details of the software and their versions.
Research Title: Hybrid Detection of Cross-Site Scripting (XSS) Vulnerability in Web Applications

2 System Details

The details of the system which is used for development have been mentioned below:

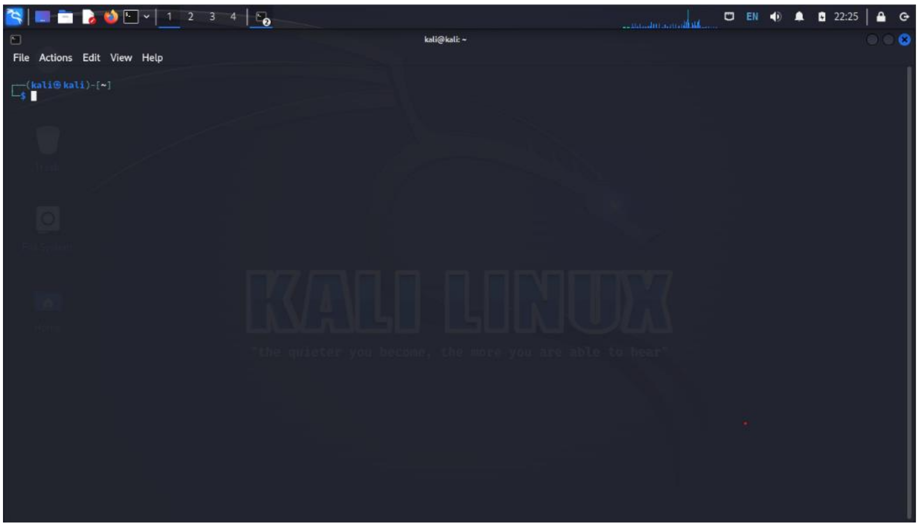
Feature	Description
Operating System	Windows 11
System information	Acer Nitro 5
Processor	i7 – 10th Gen
Memory	500MB

3 System Configuration

The proposed tool can be executed on windows and kali Linux. For executing this proposed tool in Kali Linux, the below steps can be followed.

1. Kali Linux Terminal - Download the zip file, unzip the file, and enter the tool directory. Also, the tool is uploaded in GitHub[1].

    - cd XSSFind



```
┌──(kali㉿kali)-[~]
└─$ 
```

kali@kali: ~
File Actions Edit View Help
KALI LINUX
"the quieter you become, the more you are able to hear"

[1] https://github.com/Yog267/XSSFind

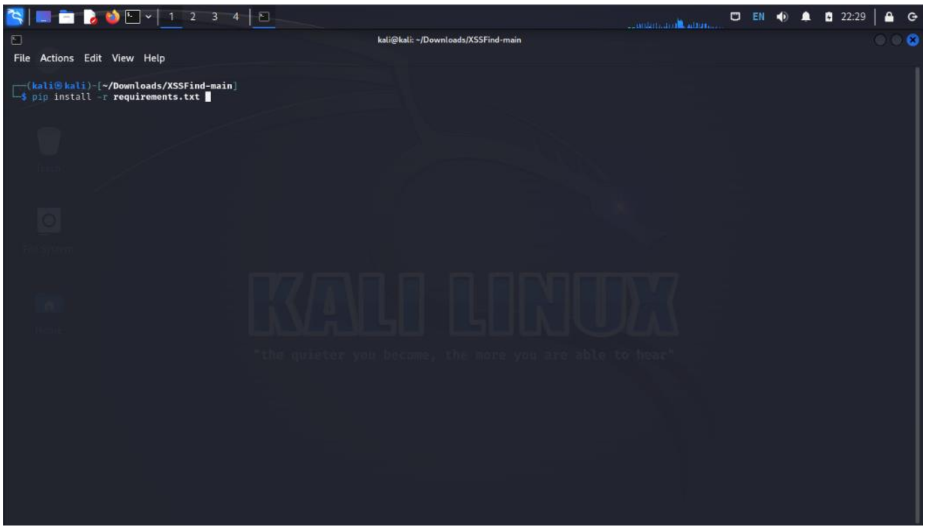
Figure 1 – Kali terminal

2. Create new virtual environment and activate with the below code:

- python -m venv venv
- source venv/bin/activate

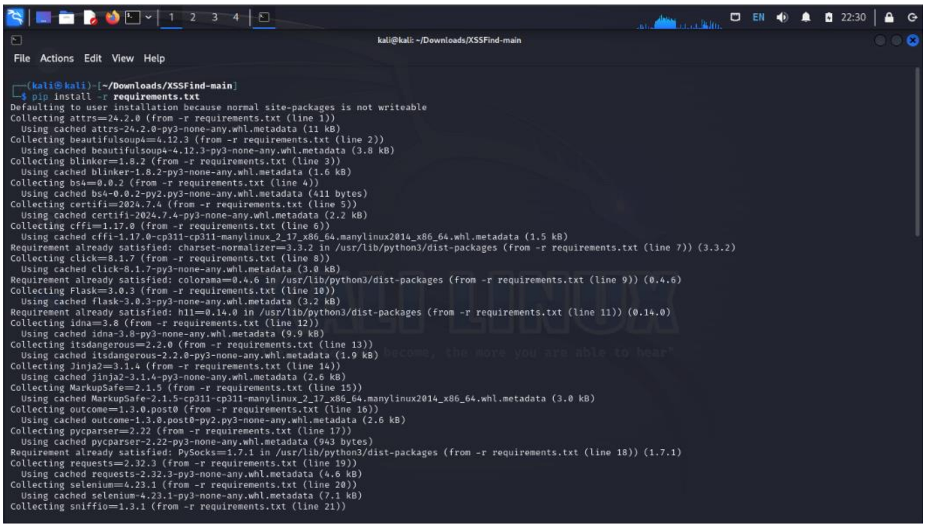
3. There are some prerequisites that has to be installed in the system to run the tool. The tool folder has requirement.txt file which has to be used to install them. Below code has to be executed to install the dependencies.

- pip install -r requirements.txt



```
┌──(kali㉿kali)-[~/Downloads/XSSFind-main]
└─$ pip install -r requirements.txt
```

Figure 2: Installing the required prerequisites

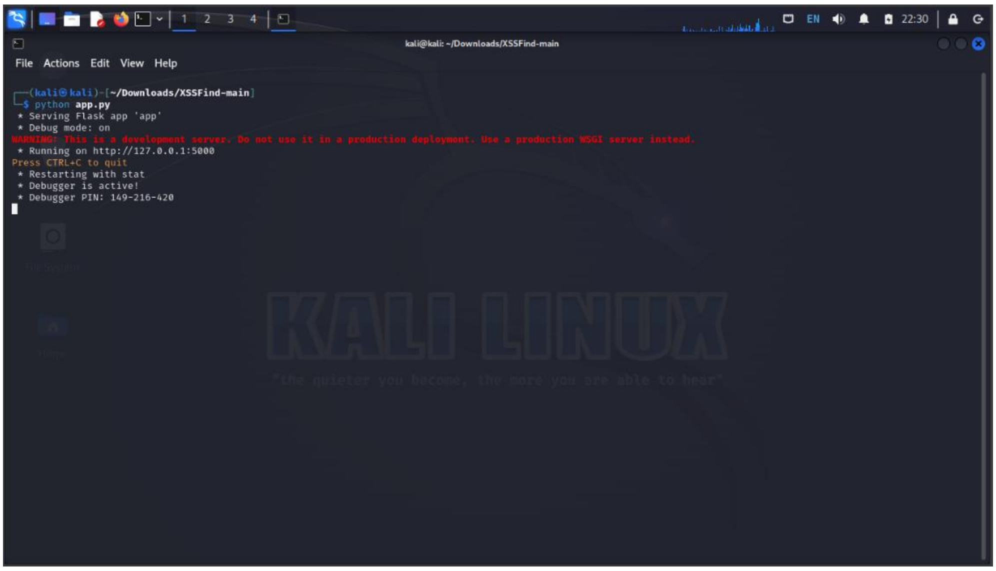


```
┌──(kali㉿kali)-[~/Downloads/XSSFind-main]
└─$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting attrs==24.2.0 (from -r requirements.txt (line 1))
  Using cached attrs-24.2.0-py3-none-any.whl.metadata (11 kB)
Collecting beautifulsoup4==4.12.3 (from -r requirements.txt (line 2))
  Using cached beautifulsoup4-4.12.3-py3-none-any.whl.metadata (3.8 kB)
Collecting blinker==1.8.2 (from -r requirements.txt (line 3))
  Using cached blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting bs4==0.0.2 (from -r requirements.txt (line 4))
  Using cached bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Collecting certifi==2024.7.4 (from -r requirements.txt (line 5))
  Using cached certifi-2024.7.4-py3-none-any.whl.metadata (2.2 kB)
Collecting cffi==1.17.0 (from -r requirements.txt (line 6))
  Using cached cffi-1.17.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: charset-normalizer==3.3.2 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 7)) (3.3.2)
Collecting click==8.1.7 (from -r requirements.txt (line 8))
  Using cached click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: colorama==0.4.6 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 9)) (0.4.6)
Collecting Flask==3.0.3 (from -r requirements.txt (line 10))
  Using cached flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: h11==0.14.0 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 11)) (0.14.0)
Collecting idna==3.8 (from -r requirements.txt (line 12))
  Using cached idna-3.8-py3-none-any.whl.metadata (9.9 kB)
Collecting itsdangerous==2.2.0 (from -r requirements.txt (line 13))
  Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting Jinja2==3.1.4 (from -r requirements.txt (line 14))
  Using cached jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting MarkupSafe==2.1.5 (from -r requirements.txt (line 15))
  Using cached MarkupSafe-2.1.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.0 kB)
Collecting outcome==1.3.0.post0 (from -r requirements.txt (line 16))
  Using cached outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)
Collecting pycparser==2.22 (from -r requirements.txt (line 17))
  Using cached pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Requirement already satisfied: PySocks==1.7.1 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 18)) (1.7.1)
Collecting requests==2.32.3 (from -r requirements.txt (line 19))
  Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting selenium==4.23.1 (from -r requirements.txt (line 20))
  Using cached selenium-4.23.1-py3-none-any.whl.metadata (7.1 kB)
Collecting sniffio==1.3.1 (from -r requirements.txt (line 21))
```

Figure 3: Installation of prerequisites

4. To run the flask application the app.py code must be executed, where the application is hosted in the localhost port 5000. The below command initiates the web UI using flask,

- python app,py



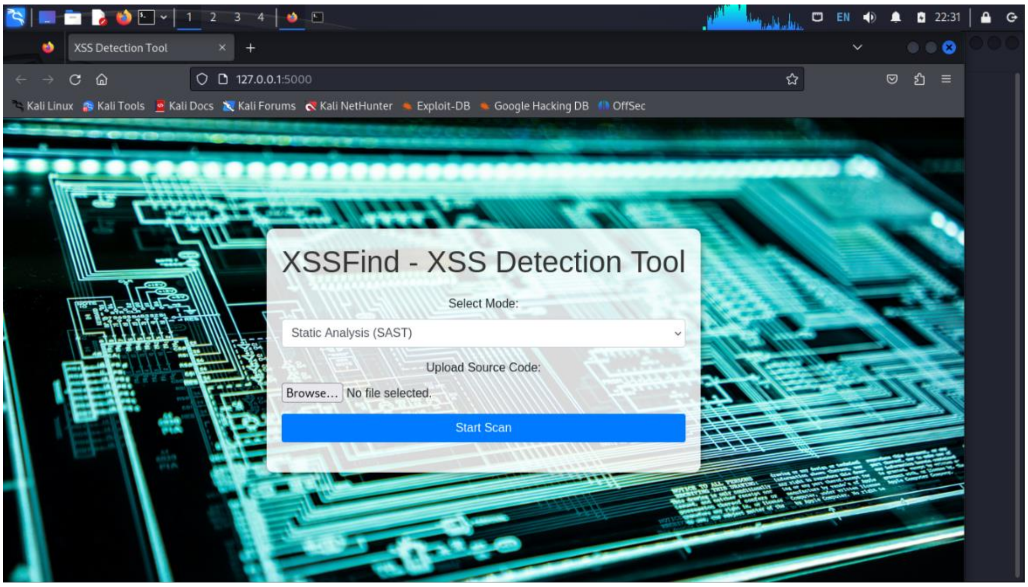
kali@kali: ~/Downloads/XSSFind-main
File Actions Edit View Help
KALI LINUX
"the quieter you become, the more you are able to hear"

```
┌──(kali㉿kali)-[~/Downloads/XSSFind-main]
└─$ python app.py
 * Serving Flask app 'app'
 * 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
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 149-216-420
```

Figure 4: Executing app.py

4 Website interface

- The interface has a selection of three options SAST analysis, DAST analysis and Hybrid analysis.

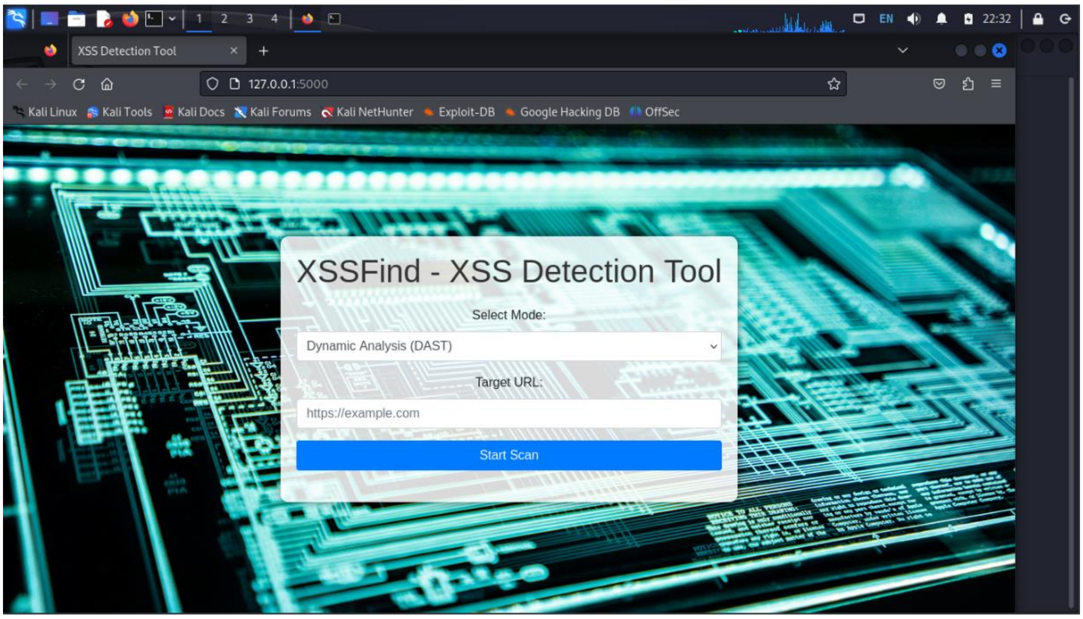


Select Mode: Static Analysis (SAST)

Upload Source Code: No file selected.

Figure 5: Interface

- If the Dynamic Analysis is chosen, then Target URL field will be available to add the website URL for analysis.



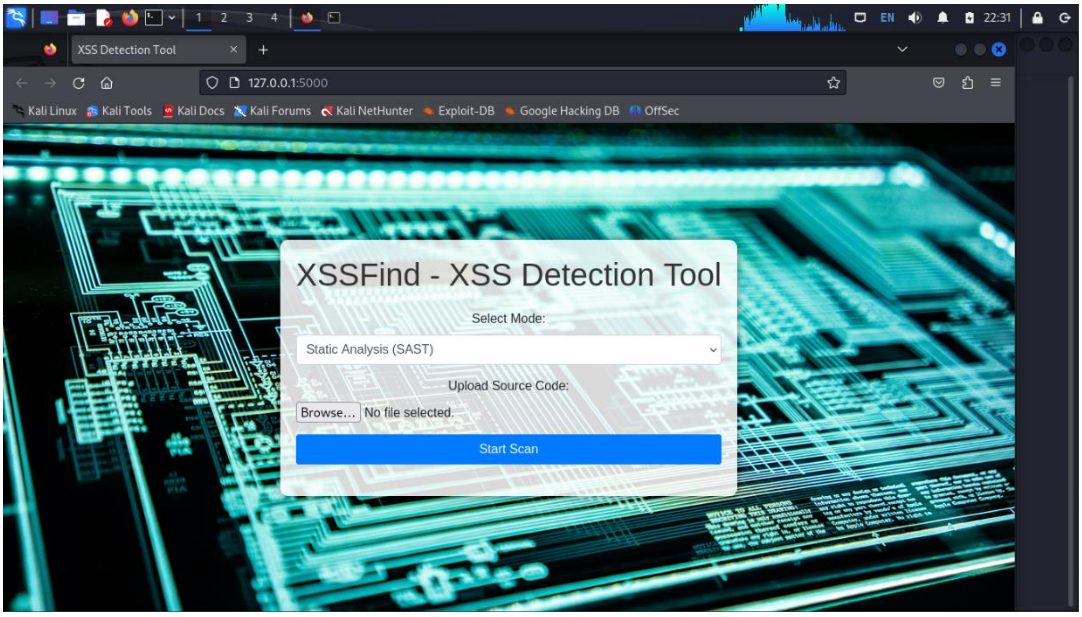
XSSFind - XSS Detection Tool

Select Mode: Dynamic Analysis (DAST)

Target URL: https://example.com

Figure 6: Dynamic Analysis

- If Static Analysis is chosen, then file upload option will be available to upload source code for analysis.



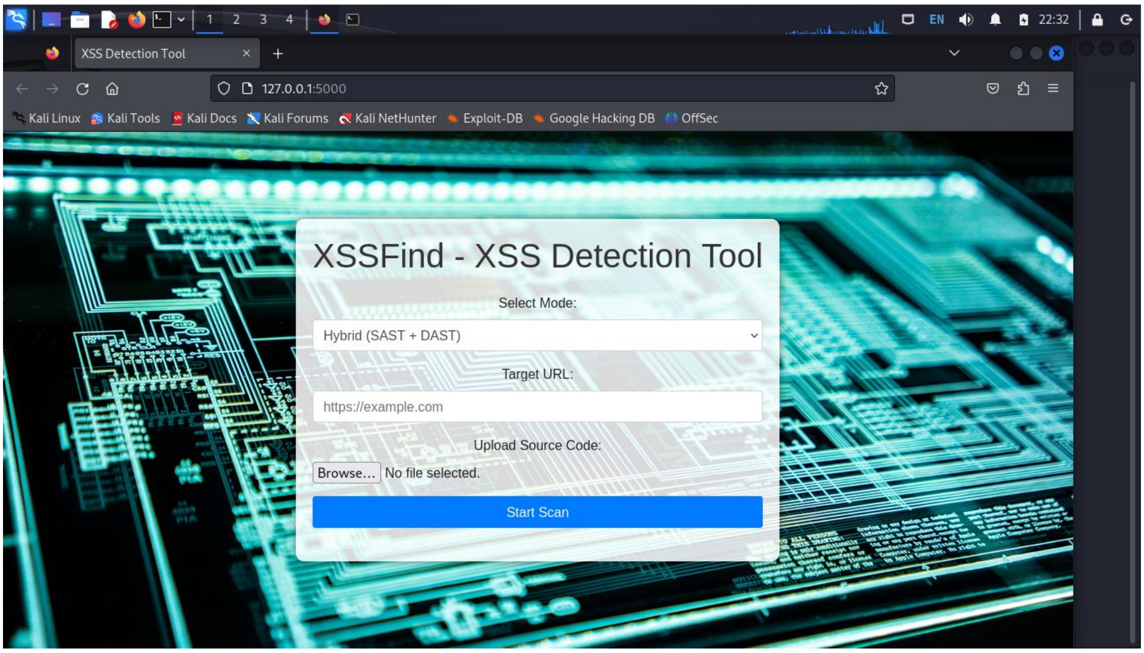
XSSFind - XSS Detection Tool

Select Mode: Static Analysis (SAST)

Upload Source Code: No file selected.

Figure 7: Static Analysis

- If Hybrid analysis is chosen it will give both the URL field and File upload option, which analyses both.



XSSFind - XSS Detection Tool

Select Mode: Hybrid (SAST + DAST)

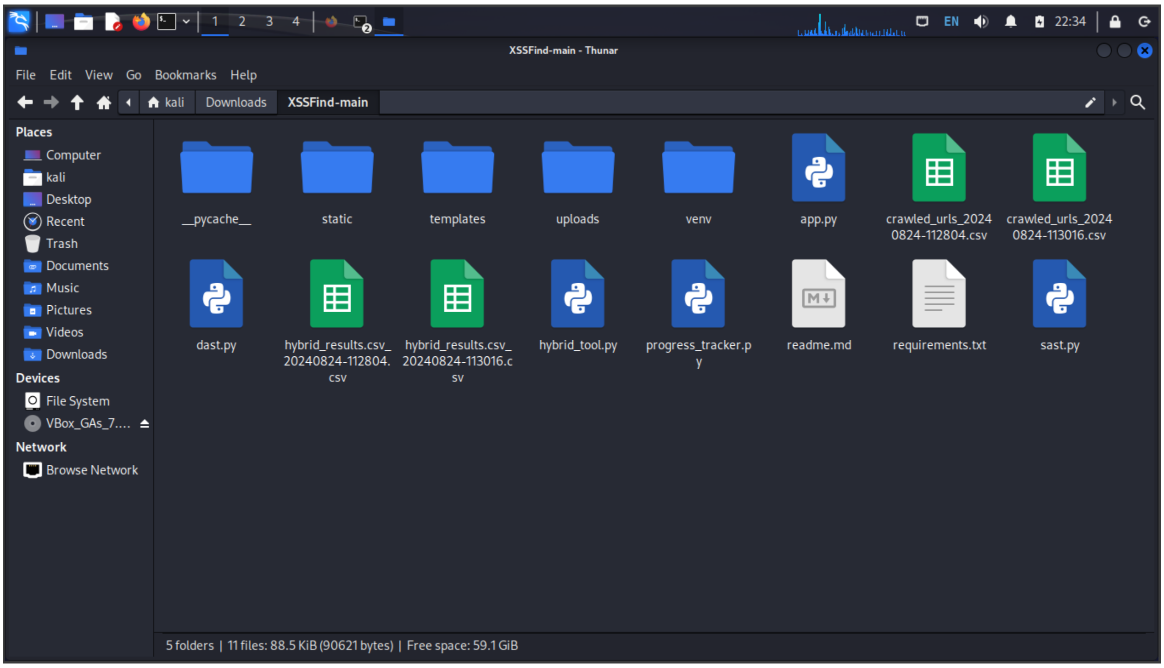
Target URL: https://example.com

Upload Source Code: Browse… No file selected.

Start Scan

Figure 8: Hybrid analysis

- Results will be stored automatically in the form of .csv file in the same tool directory.



__pycache__ static templates uploads venv app.py crawled_urls_20240824-112804.csv crawled_urls_20240824-113016.csv dast.py hybrid_results.csv_20240824-112804.csv hybrid_results.csv_20240824-113016.csv hybrid_tool.py progress_tracker.py readme.md requirements.txt sast.py

5 folders | 11 files: 88.5 KiB (90621 bytes) | Free space: 59.1 GiB

Figure 9: Results