Project Summary

Python-Based Web Crawler with Port Scanner and Vulnerability Scanner

1. Introduction

This capstone project delivers a Python-based GUI security tool that integrates three essential functions of reconnaissance and vulnerability management: web crawling, port scanning, and vulnerability detection. The primary objective was to design a modular, GUI-based application that demonstrates how widely used open-source tools can be orchestrated to create a practical and educational vulnerability assessment platform.

2. System Modules

The project is organized into three core modules, each implemented in Python and tested individually before being integrated into a Tkinter-based GUI:

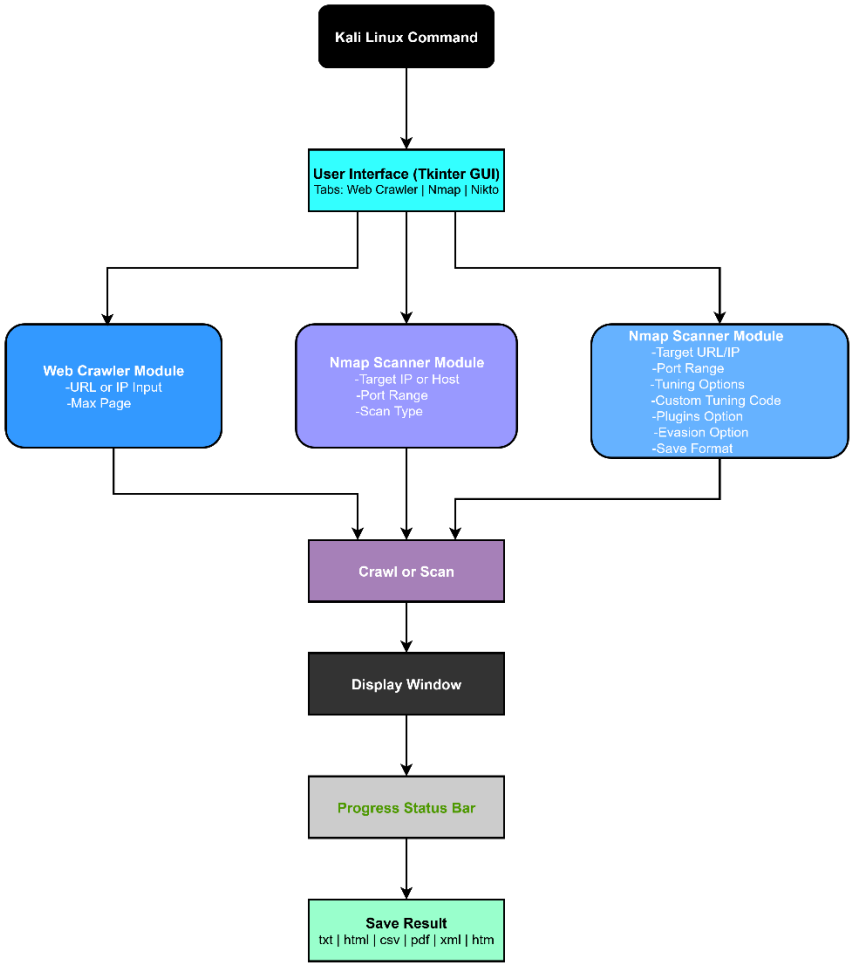

Fig 1: Architectural Diagram of Python-Based Web Crawler and Vulnerability Scanner

2.1 Graphical User Interface (GUI)

The Tkinter-based GUI separates each module into its own tab for clarity and ease of use. Long-running scans execute in background threads to keep the interface responsive. Features include:

- Real-time scrolling output window.
- Green progress bar for scan activity.
- Status messages that update dynamically.
- Cancel button to terminate scans safely.
- Export functionality for saving results in multiple formats.

2.2 Web Crawler

- Built using requests and BeautifulSoup.
- Crawls permitted targets, extracts titles, metadata, and discovered links.
- Outputs structured records suitable for manual inspection or extended analysis.

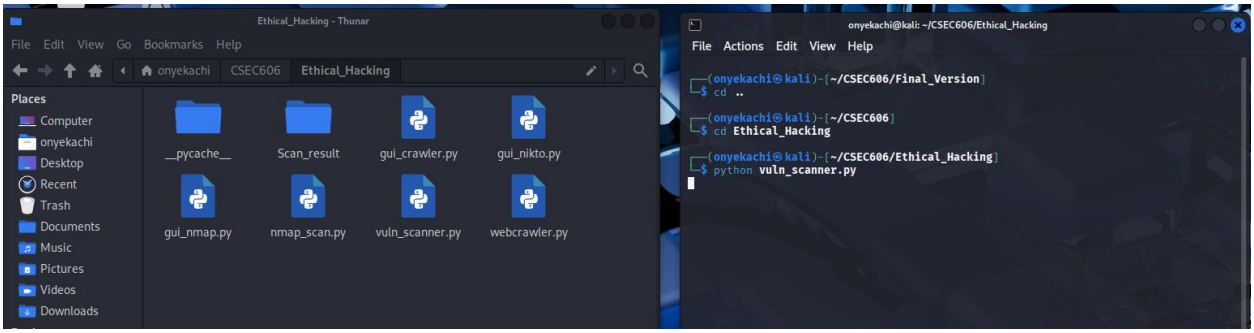

Fig 2: Folder Overview & Kali Command Line

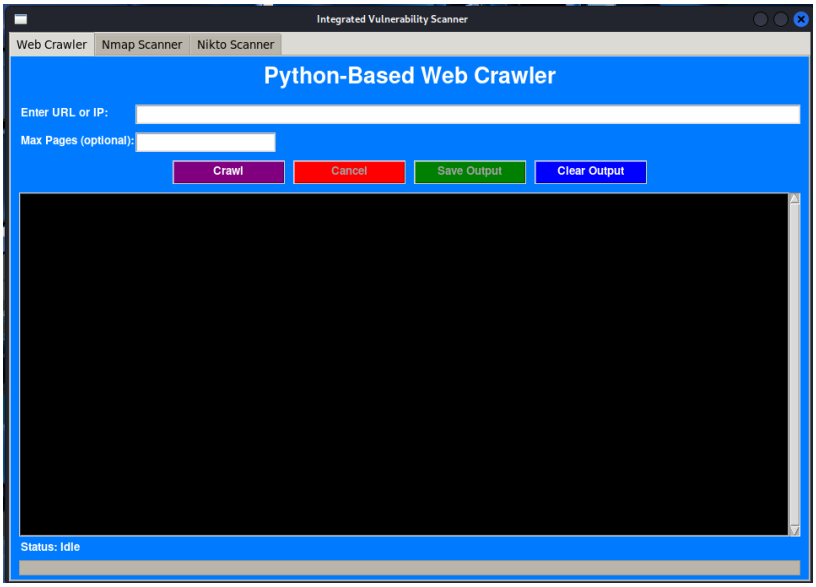

Fig 3: Web Crawler GUI

2.3 Nmap Port Scanner

- Wraps the native nmap tool using Python's subprocess.
- Offers three scanning profiles: SYN Scan, SYN+OS+Version detection, and Aggressive Scan.
- Allows port range customization for flexible discovery depth.

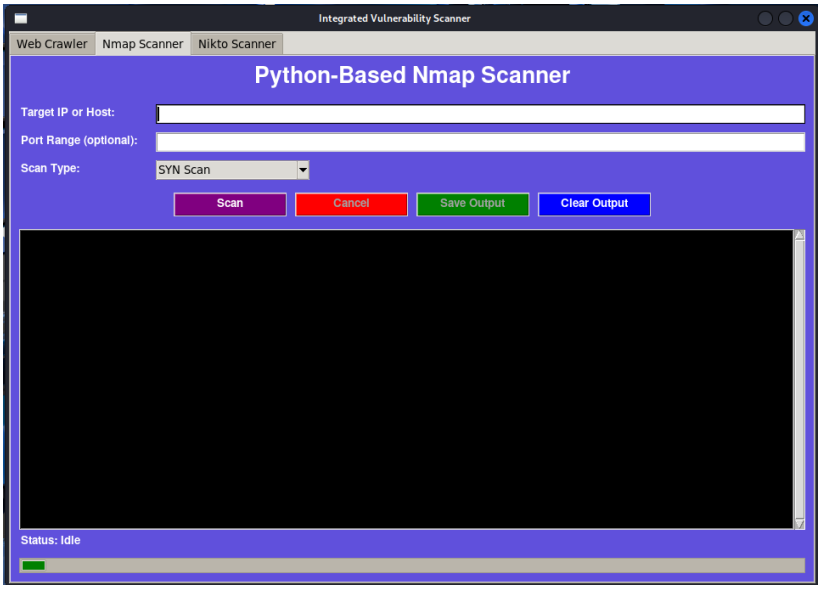

Fig 4A: Nmap Scanner GUI

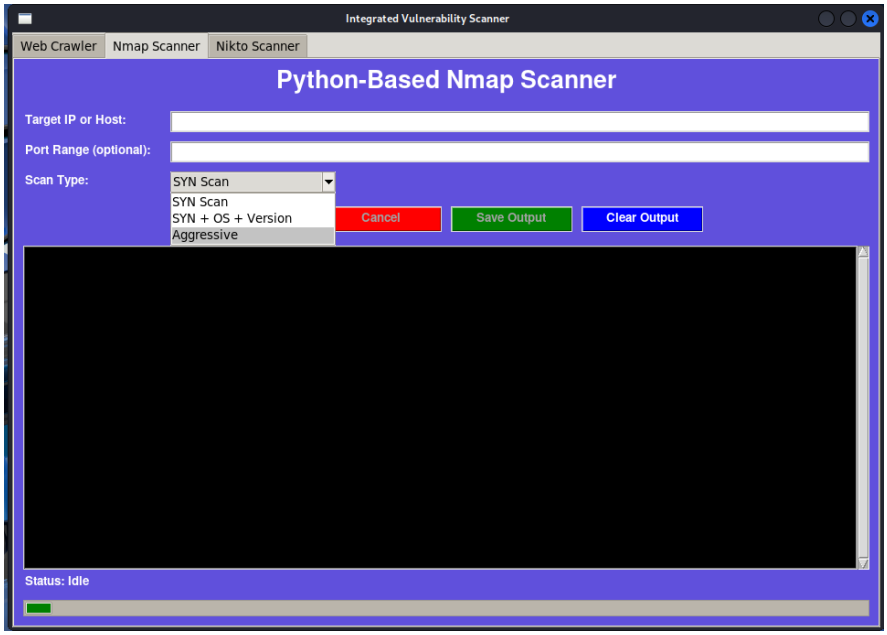

Fig 4B: Web Crawler GUI Scan Type Options

2.4 Nikto Vulnerability Scanner

- Integrates Nikto for HTTP vulnerability checks.
- Exposes tuning codes and custom payloads for advanced scanning.
- Preserves HTML reports while also enabling export to TXT, CSV, and PDF formats.

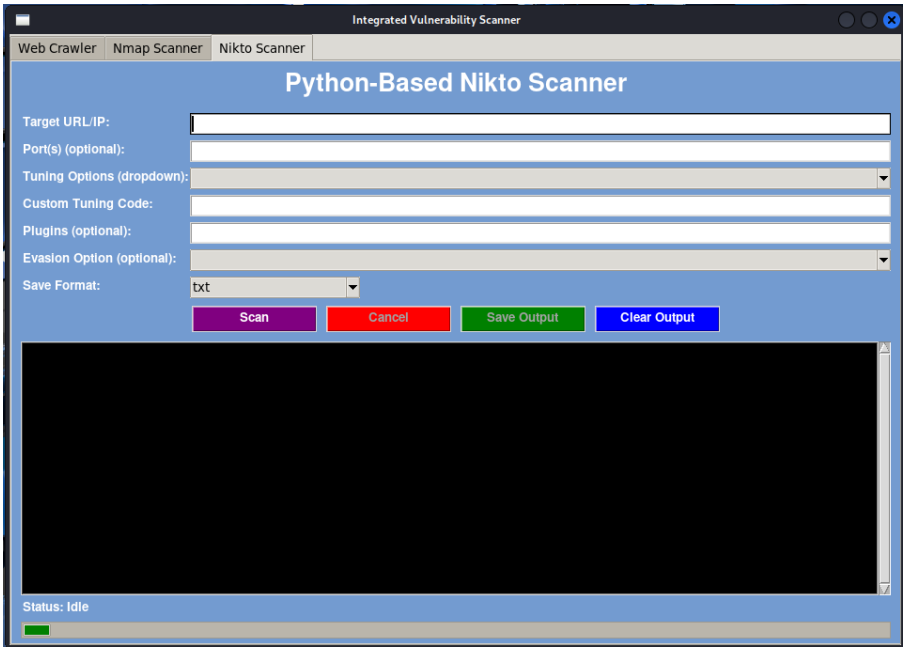

Fig 5A: Nikto Scanner GUI

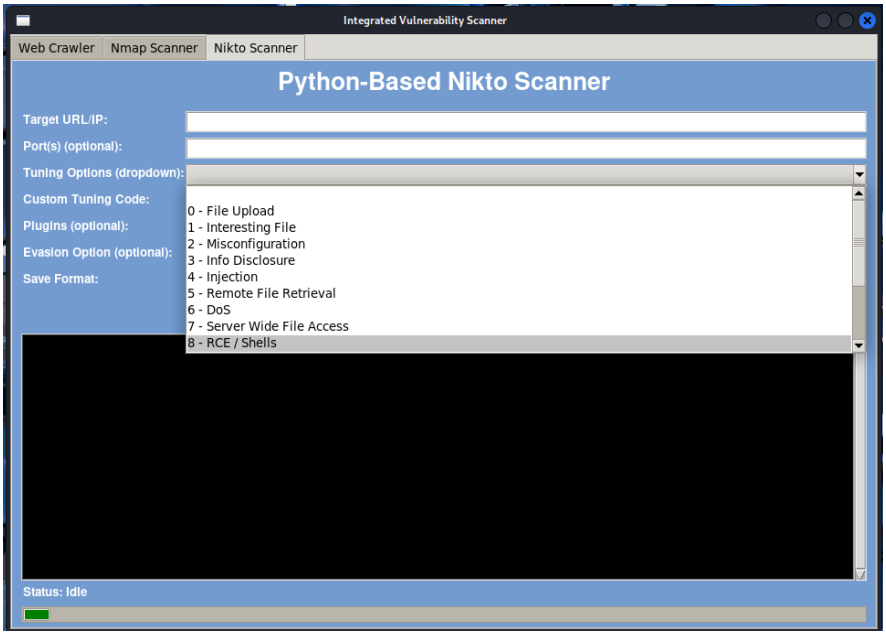

Fig 5B: Nikto Scanner GUI Tuning Options

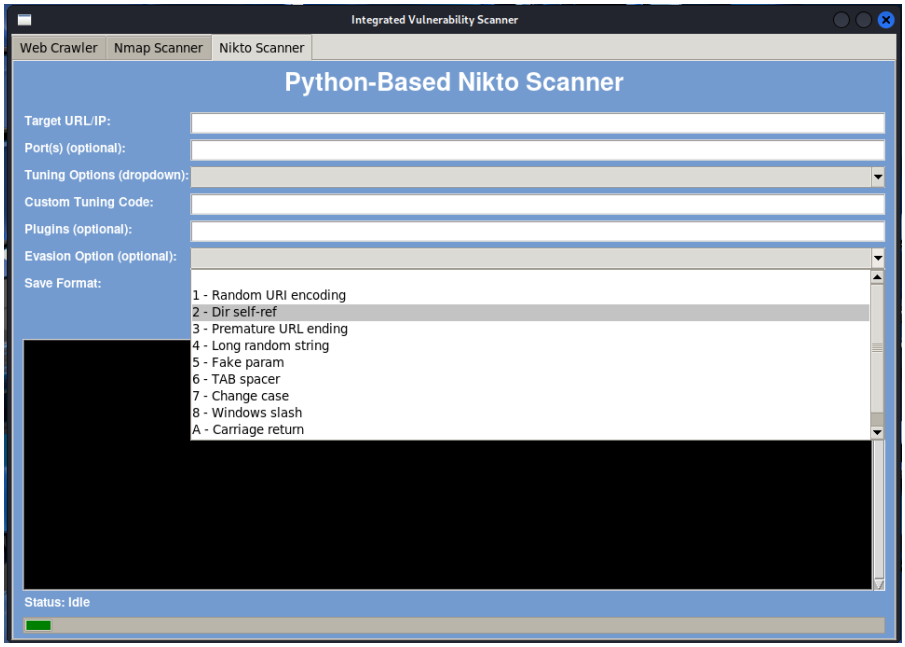

Fig 5C: Nikto Scanner GUI Evasion Options

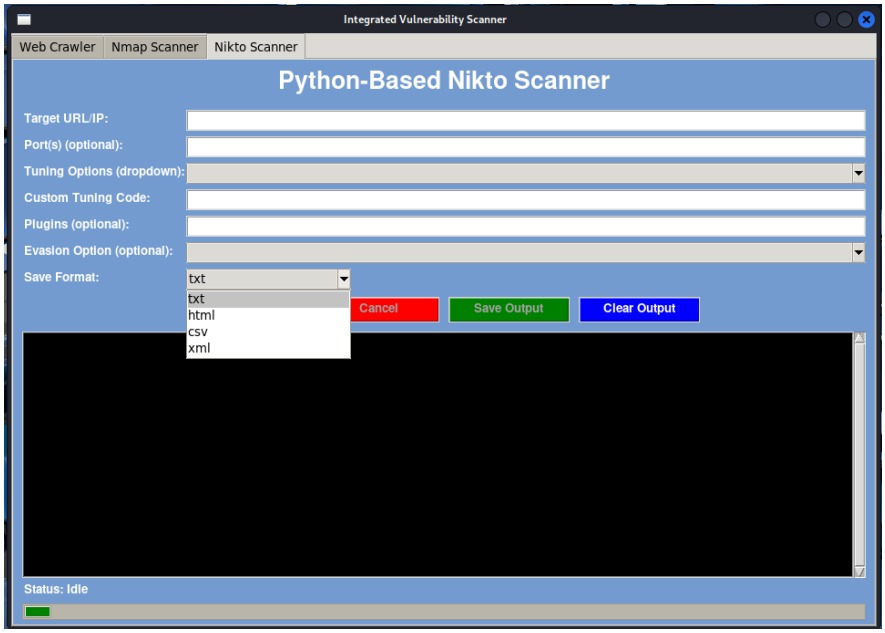

Fig 5D: Nikto Scanner GUI Save Format Options

3. Testing and Validation

All scans and experiments were conducted in a controlled lab environment using Kali Linux as the host and Metasploitable VM as the target. Key results include:

- The Nmap module produced identical results to standalone Nmap runs.
- Nikto vulnerability findings were accurately reported and successfully exported.
- The Cancel function reliably stopped subprocesses without freezing the GUI.
- Progress and status indicators matched actual scanner behavior.

5. Ethical and Legal Considerations

The project emphasizes ethical use and includes explicit reminders within the interface and documentation. Testing was strictly limited to lab environments. Users are cautioned that unauthorized scanning of external systems is illegal and may violate cybersecurity laws. The tool is provided strictly for educational, research, and demonstration purposes.

6. Future Enhancements

Planned improvements to extend the system's capabilities include:

- Scheduling scans and archiving results for historical analysis.
- Database integration to store and query scan history.
- Enrichment with CVE databases for contextual vulnerability details.
- Migration to a hybrid model where lightweight orchestration is handled by serverless functions, and heavy scans run on ephemeral cloud-based virtual machines.

7. Conclusion

This Python-based vulnerability management tool demonstrates how separate reconnaissance and scanning utilities can be unified in a single, responsive GUI. The modular design ensures clarity and maintainability, while the integrated architecture provides a strong foundation for future cloud-enabled automation. The project highlights both the technical feasibility of such integration and the ethical responsibility required when applying security tools.