Vigilant Core OWASP Vulnerability and Phishing Defense Platform

K. Rani
Assistant Professor
Department of IT
K.L.N. College of Engineering,
Sivagangai, India

M. K. Baruni
UG Scholar
Department of IT
K.L.N. College of Engineering,
Sivagangai, India

N. R. G. Laksitha
UG Scholar
Department of IT
K.L.N. College of Engineering,
Sivagangai, India

***Abstract* - This paper presents the design and implementation of a web security tool that combines phishing website detection and basic web vulnerability scanning. The system is developed using Python and integrates custom phishing detection logic, URL analysis, and web scraping techniques to identify malicious websites. The proposed system analyzes various features such as URL length, presence of IP address in URL, HTTPS usage, suspicious keywords, redirection behavior, and HTML form actions to determine whether a website is phishing or legitimate. In addition to phishing detection, the tool also performs basic vulnerability scanning by analyzing web page content, open ports, and security misconfigurations. The system includes a graphical user interface to provide user-friendly interaction and real-time scanning results. The experimental results show that the system can effectively identify phishing websites and detect common web vulnerabilities. This tool can be useful for users, students, and small organizations to improve awareness and protection against phishing attacks and web security threats.**

Keywords: Phishing Detection, Web Vulnerability Scanner, URL Analysis, Cyber Security, Python, Web Security, Threat Detection.

I. INTRODUCTION

The rapid growth of internet services and online transactions has significantly increased the number of cyber security threats, particularly phishing attacks and web application vulnerabilities. Phishing websites are designed to impersonate legitimate websites in order to steal sensitive information such as login credentials, banking details, and personal data. At the same time, web applications often contain vulnerabilities that can be exploited by attackers to gain unauthorized access or disrupt services. Therefore, detecting phishing websites and identifying web vulnerabilities has become an important area of research in cyber security.

Traditional phishing detection systems relied mainly on blacklist-based approaches, where known phishing URLs were stored in databases and compared with user-requested URLs. These systems were effective in detecting previously reported phishing websites but failed to identify newly created phishing sites and zero-day attacks. Additionally, blacklist systems required continuous updates and large databases, which reduced detection efficiency and increased system overhead [1], [4].

To overcome these limitations, heuristic and rule-based phishing detection techniques were introduced. These systems analyzed various URL features such as URL length, presence of special characters, use of IP address instead of domain name, and HTTPS protocol usage to determine whether a website was suspicious. These approaches improved phishing detection accuracy but were limited to URL-based analysis and did not consider webpage content or form structures [5], [14].

With advancements in web scraping and content analysis, content-based phishing detection methods were developed to analyze HTML structure, login forms, external links, and redirection behavior. These systems provided better detection accuracy by examining webpage content instead of relying only on URL features. However, most of these systems were developed as standalone phishing detection tools without integrating web vulnerability scanning capabilities [3], [6].

Meanwhile, web vulnerability scanning tools were developed to identify security weaknesses such as SQL injection, cross-site scripting, insecure HTTP headers, and server misconfigurations. These tools helped developers and security analysts identify vulnerabilities in web applications and improve system security. However, many vulnerability scanners were complex, command-line based, and required advanced technical knowledge to operate [8], [9].

Security organizations and research communities also introduced standardized vulnerability classifications such as the OWASP Top 10, which lists the most critical web application security risks. These standards helped developers understand common vulnerabilities and implement secure coding practices, but they did not provide integrated phishing detection mechanisms within the same platform [7], [11].

Recent research efforts have focused on integrating multiple cyber security functionalities such as phishing detection, vulnerability scanning, and security reporting into a single platform. These integrated systems improve security assessment efficiency and provide centralized monitoring and analysis. However, many existing systems lack user-friendly graphical interfaces and customizable phishing detection logic suitable for educational and small-scale security analysis environments [10], [13].

Therefore, this project proposes a **Phishing Website Detection and Web Vulnerability Scanner** that integrates custom phishing detection logic, URL analysis, web scraping, and basic vulnerability scanning into a single user-friendly system. The proposed system aims to detect phishing websites, identify common web vulnerabilities, reduce false positives, and provide an automated security analysis tool for users, students, and researchers.

Machine learning and intelligent phishing detection systems were later developed to improve detection accuracy and reduce false positives. These systems used classification algorithms, feature extraction, and pattern analysis to identify phishing websites more efficiently than traditional rule-based systems. Similarly, modern web vulnerability scanning tools were developed to automatically detect security weaknesses in web applications and generate security reports. These tools improved vulnerability detection speed and accuracy, but many of them were commercial tools and not suitable for educational or research-based environments [2], [12], [15].

II. METHODOLOGY

The proposed Phishing Website Detection and Web Vulnerability Scanner is designed as an integrated security analysis system that detects phishing websites and identifies basic web application vulnerabilities. The system workflow begins with URL input and proceeds through URL analysis, phishing detection, web content analysis, vulnerability scanning, and report generation, forming a complete website security analysis pipeline.

A. Requirement Analysis and System Design

The initial phase of the proposed Phishing Website Detection and Web Vulnerability Scanner involved identifying the core functional and system requirements necessary for effective security analysis. The primary modules defined include URL input and validation, phishing detection using custom logic, web vulnerability scanning, content analysis through web scraping, and report generation. These modules collectively ensure a complete end-to-end website security assessment process.

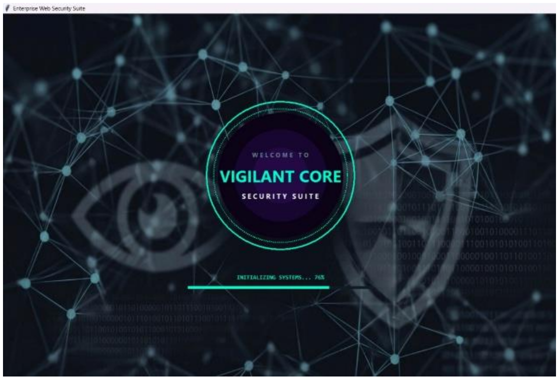


Figure 1: Initialization Screen of the Vigilant Core Security Suite

B. Frontend Development Using Graphical User Interface (GUI)

The frontend of the proposed Phishing Website Detection and Web Vulnerability Scanner is developed using a Python-based graphical user interface to ensure simplicity and ease of use. The interface provides a centralized dashboard where users can access key modules such as the Web Vulnerability Scanner and Phishing Defense Center. The design focuses on clear navigation, interactive controls, and real-time result display. The theme module defines the visual layout, color scheme, and interface components. GUI elements such as buttons, input fields, and result panels enable smooth user interaction, while event handling ensures efficient execution of scanning operations. This design allows users to perform security analysis easily without requiring advanced technical knowledge.

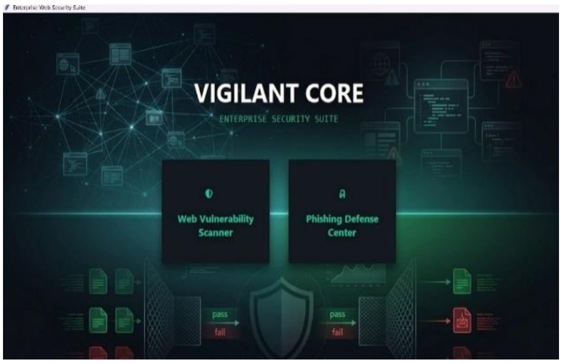

Enterprise Web Security Suite
VIGILANT CORE
ENTERPRISE SECURITY SUITE
Web Vulnerability Scanner
Phishing Defense Center
pass
fail
pass
fail

Figure 2: Main Interface of the Vigilant Core Security Suite Showing Security Modules

C. Backend Integration and Processing

The backend of the proposed system is implemented using Python-based modules that perform URL analysis, phishing detection, and web vulnerability scanning. It integrates multiple libraries to handle network requests, data extraction, and real-time security analysis. The scanning process is managed through backend logic supporting Start, Pause, and Stop functionalities, enabling flexible control of operations. The system processes user-input URLs and continuously updates the interface with live results. It also includes output management features such as Copy Output, Clear Terminal, and HTML report generation for downloading detailed scan results. This integration ensures efficient data processing, real-time updates, and reliable report generation for effective security analysis. The backend architecture is designed to be modular, allowing easy integration of additional security features in the future.

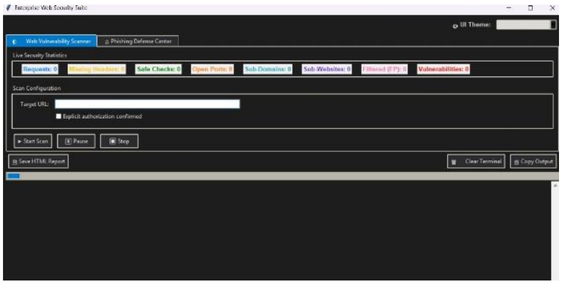
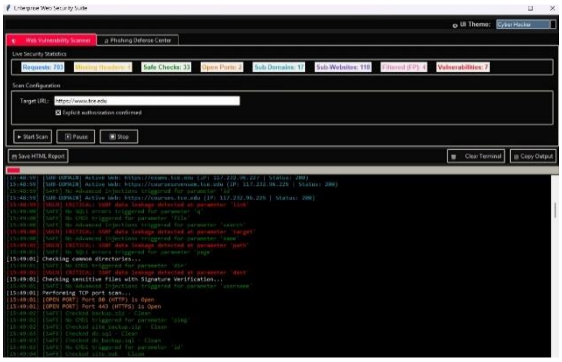

Enterprise Web Security Suite
UI Theme:
Web Vulnerability Scanner | Phishing Defense Center
Live Security Statistics
Requests: 0 | Missing Headers: 0 | Safe Checks: 0 | Open Ports: 0 | Sub-Domains: 0 | Sub-Websites: 0 | Filtered (FP): 0 | Vulnerabilities: 0
Scan Configuration
Target URL:
□ Explicit authorization confirmed

Figure 3: Web Vulnerability Scanner Interface with Scan Controls and Security Metrics

D. Vulnerability Scanning & Report Generation

The proposed system performs automated scanning of websites based on user-provided URLs entered in the target URL field. Before initiating the scan, users must confirm explicit authorization by selecting the provided checkbox, ensuring ethical and permitted usage. The scanning process is then started using the Start button, with additional controls such as Pause and Stop allowing flexible execution of scanning tasks. The system executes multiple security checks automatically, including URL validation, port scanning (open/closed ports), vulnerability detection, and content analysis. Real-time results are continuously displayed within the interface, showing detailed logs of detected issues, safe checks, and network activity. The dashboard provides a summarized view of key metrics such as total requests, subdomains, sub-websites, open ports, filtered results (false positives), and detected vulnerabilities. This dynamic visualization helps users quickly assess the security status of the target website. Additionally, the system supports report generation by allowing users to download detailed HTML reports of the scanning results. These reports include vulnerability details, analysis summaries, and scan logs for further review and documentation. This automated process ensures efficient security analysis, accurate detection of threats, and easy access to comprehensive security reports.

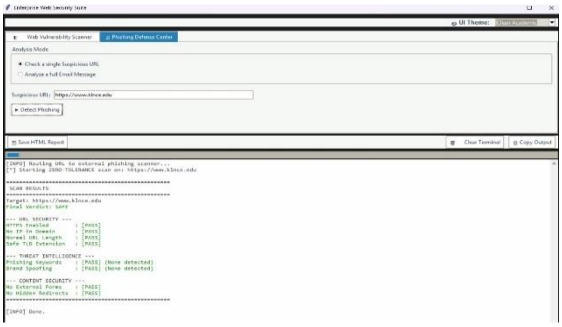

Enterprise Web Security Suite
UI Theme: Cyber Hacker
Web Vulnerability Scanner | Phishing Defense Center
Live Security Statistics
Requests: 703 | Missing Headers: 4 | Safe Checks: 33 | Open Ports: 2 | Sub-Domains: 17 | Sub-Websites: 118 | Filtered (FP): 4 | Vulnerabilities: 7
Scan Configuration
Target URL: https://www.tce.edu
☒ Explicit authorization confirmed

```
[15:48:59] [SUB-DOMAIN] Active Web: https://exams.tce.edu (IP: 117.232.96.227 | Status: 200)
[15:48:59] [SUB-DOMAIN] Active Web: https://coursesevensem.tce.edu (IP: 117.232.96.225 | Status: 200)
[15:48:59] [SAFE] No Advanced Injections triggered for parameter 'id'
[15:48:59] [SUB-DOMAIN] Active Web: https://courses.tce.edu (IP: 117.232.96.225 | Status: 200)
[15:48:59] [VULN] CRITICAL: SSRF data leakage detected at parameter 'link'
[15:49:00] [SAFE] No SQLi errors triggered for parameter 'q'
[15:49:00] [SAFE] No CMDi triggered for parameter 'file'
[15:49:00] [SAFE] No Advanced Injections triggered for parameter 'search'
[15:49:00] [VULN] CRITICAL: SSRF data leakage detected at parameter 'target'
[15:49:00] [SAFE] No Advanced Injections triggered for parameter 'name'
[15:49:01] [VULN] CRITICAL: SSRF data leakage detected at parameter 'path'
[15:49:01] [SAFE] No SQLi errors triggered for parameter 'page'
[15:49:01] Checking common directories...
[15:49:01] [SAFE] No CMDi triggered for parameter 'dir'
[15:49:01] [VULN] CRITICAL: SSRF data leakage detected at parameter 'dest'
[15:49:01] Checking sensitive files with Signature Verification...
[15:49:01] [SAFE] No Advanced Injections triggered for parameter 'username'
[15:49:01] Performing TCP port scan...
[15:49:01] [OPEN PORT] Port 80 (HTTP) is Open
[15:49:01] [OPEN PORT] Port 443 (HTTPS) is Open
[15:49:02] [SAFE] Checked backup.zip - Clean
[15:49:02] [SAFE] No CMDi triggered for parameter 'ping'
[15:49:02] [SAFE] Checked site_backup.zip - Clean
[15:49:03] [SAFE] Checked db.sql - Clean
[15:49:03] [SAFE] Checked db_backup.sql - Clean
[15:49:03] [SAFE] No CMDi triggered for parameter 'id'
[15:49:04] [SAFE] Checked site.bak - Clean
```

Figure 4: Web Vulnerability Scan Results Display with Real-Time Analysis Output

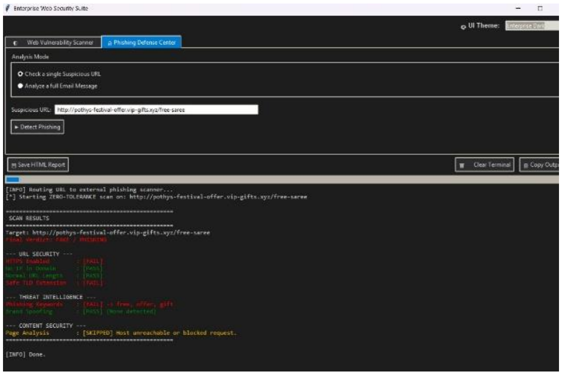
E. URL & Mail based Phishing Detection Process

The Phishing Defense Center interface used to analyze suspicious URLs for potential phishing threats. The user selects the analysis mode, such as checking a single suspicious URL or analyzing an email message, and enters the target URL into the input field. Once the Detect Phishing action is initiated, the system performs a comprehensive security evaluation. The output panel displays the final verdict along with detailed checks, including URL security, HTTPS status, domain validation, phishing keyword detection, brand spoofing analysis, and content security verification


Enterprise Web Security Suite
UI Theme: Clean Academic
Web Vulnerability Scanner | Phishing Defense Center
Analysis Mode
● Check a single Suspicious URL
○ Analyze a full Email Message
Suspicious URL: https://www.klnce.edu

```
[INFO] Routing URL to external phishing scanner...
[*] Starting ZERO-TOLERANCE scan on: https://www.klnce.edu

==================================================
 SCAN RESULTS
==================================================
Target: https://www.klnce.edu
Final Verdict: SAFE

--- URL SECURITY ---
HTTPS Enabled        : [PASS]
No IP in Domain      : [PASS]
Normal URL Length    : [PASS]
Safe TLD Extension   : [PASS]

--- THREAT INTELLIGENCE ---
Phishing Keywords    : [PASS] (None detected)
Brand Spoofing       : [PASS] (None detected)

--- CONTENT SECURITY ---
No External Forms    : [PASS]
No Hidden Redirects  : [PASS]
==================================================

[INFO] Done.
```

Figure 5: Phishing Detection Result Showing Safe URL Classification with Security Checks Passed

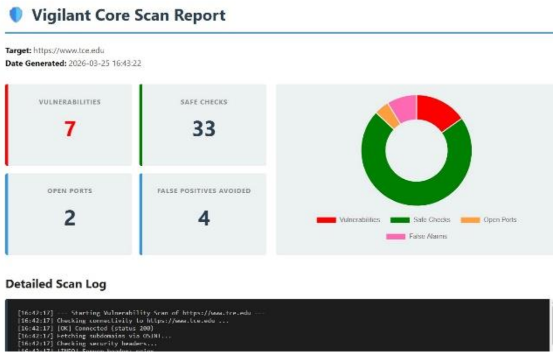

Enterprise Web Security Suite
UI Theme: Enterprise Dark
Web Vulnerability Scanner | Phishing Defense Center
Analysis Mode
○ Check a single Suspicious URL
● Analyze a full Email Message
Suspicious URL: http://pothys-festival-offer.vip-gifts.xyz/free-saree

```
[INFO] Routing URL to external phishing scanner...
[*] Starting ZERO-TOLERANCE scan on: http://pothys-festival-offer.vip-gifts.xyz/free-saree

==================================================
 SCAN RESULTS
==================================================
Target: http://pothys-festival-offer.vip-gifts.xyz/free-saree
Final Verdict: FAKE / PHISHING

--- URL SECURITY ---
HTTPS Enabled        : [FAIL]
No IP in Domain      : [PASS]
Normal URL Length    : [PASS]
Safe TLD Extension   : [FAIL]

--- THREAT INTELLIGENCE ---
Phishing Keywords    : [FAIL] -> free, offer, gift
Brand Spoofing       : [PASS] (None detected)

--- CONTENT SECURITY ---
Page Analysis        : [SKIPPED] Host unreachable or blocked request.
==================================================

[INFO] Done.
```

Figure 6: Phishing Detection Result Showing Fake URL Classification with Detected Threat Indicators

F. HTML Report Generation and Visualization

The system provides a feature to generate and download a detailed HTML report after completing the scanning process. The report presents a structured view of the security analysis, including detected vulnerabilities, safe checks, and network-related information. The generated report reflects the same results shown in Figure 3, ensuring consistency between real-time output and stored data. It also includes graphical visualization in the form of a pie chart, representing the distribution of safe checks, vulnerabilities, and other scan metrics. Users can open and view the report in a web browser, allowing easy access to scan results even in offline mode. This functionality supports documentation, result sharing, and better understanding of security status through visual representation. The pie chart enhances visual interpretation by providing a clear summary of the overall security analysis results.



Vigilant Core Scan Report

Target: https://www.tce.edu
Date Generated: 2026-03-25 16:43:22

VULNERABILITIES 7	SAFE CHECKS 33
OPEN PORTS 2	FALSE POSITIVES AVOIDED 4

Detailed Scan Log

```
[16:42:17] --- Starting Vulnerability Scan of https://www.tce.edu ---
[16:42:17] Checking connectivity to https://www.tce.edu ...
[16:42:17] [OK] Connected (status 200)
[16:42:17] Fetching subdomains via OSINT...
[16:42:17] Checking security headers...
[16:42:17] [INFO] Server header: nginx
```

Figure 7: Downloaded HTML Report Displaying Web Security Scan Results

III. System Architecture

The proposed Phishing Website Detection and Web Vulnerability Scanner follows a modular architecture designed to ensure efficient processing, real-time analysis, and easy scalability. The system consists of three primary components: the graphical user interface (frontend), the Python-based backend processing modules, and the report generation system. These components interact to provide a seamless end-to-end website security analysis solution.

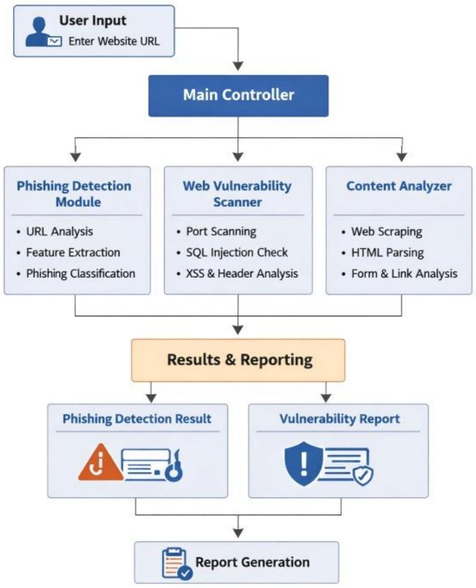


Figure 8: Overall System Architecture and Process Flow

A. Graphical User Interface Module

The Graphical User Interface (GUI) module is developed using a Python-based interface to provide a simple and interactive platform for users. It consists of dedicated sections for the Web Vulnerability Scanner and the Phishing Defense Center, enabling users to perform different security analysis tasks. This module allows users to enter target URLs, select analysis modes, and control scanning operations using Start, Pause, and Stop functionalities. It also displays real-time scanning results, system statistics, and logs, ensuring effective interaction and easy monitoring of the security analysis process.

B. Backend Layer – Python Processing Modules

The backend layer consists of Python-based modules responsible for performing core security operations. These modules handle URL analysis, phishing detection using custom logic, web scraping, and vulnerability scanning. The backend processes user inputs, performs multiple security checks such as port scanning, parameter testing, and threat detection, and continuously updates the frontend with real-time results. This layer ensures efficient processing and accurate detection of security threats.

C. Report Generation Module – HTML Reports

The system includes a report generation module that creates structured HTML reports after completing the scanning process. These reports contain detailed information such as detected vulnerabilities, safe checks, scan logs, and graphical representations like pie charts. Users can download and view these reports for documentation, sharing, and further analysis.

D. Data Flow Overview

The system workflow begins when the user enters a target URL and initiates the scanning process. The frontend sends the input to the backend modules, which perform phishing detection and vulnerability analysis. The results are processed and displayed in real time within the interface, including logs and dashboard statistics such as requests, subdomains, open ports, and vulnerabilities.

After completion, the system generates an HTML report that reflects the same results shown in the interface. This structured data flow ensures efficient communication between components, real-time result visualization, and reliable security analysis.

IV. RESULT AND DISCUSSION

The proposed Phishing Website Detection and Web Vulnerability Scanner was successfully developed and tested using a Python-based graphical user interface, custom phishing detection logic, and web vulnerability scanning modules. The system was evaluated based on functionality, performance, usability, and reliability under real-time conditions.

A. Functional Testing Results

All core modules of the system were tested to ensure proper functionality. The phishing detection module successfully analyzed URLs and classified them as safe or malicious based on multiple parameters such as HTTPS usage, domain characteristics, and suspicious patterns. The web vulnerability scanner effectively identified security issues including open ports, missing headers, subdomains, and potential vulnerabilities. The system also displayed real-time scan results, logs, and statistical summaries on the dashboard.

The system also demonstrated consistency in handling different types of input URLs, including valid, invalid, and potentially malicious links, ensuring robust functionality across various scenarios. The integration between the phishing detection module and vulnerability scanner enabled comprehensive security analysis within a single platform. Furthermore, the real-time dashboard updates improved monitoring efficiency by allowing users to track scanning progress and instantly view detected issues, enhancing the overall effectiveness of the system.

B. Performance Evaluation of the Scanning System

The system maintained stable performance even when processing multiple scanning requests sequentially, demonstrating its capability to handle continuous operations without degradation. Efficient resource utilization ensured that scanning tasks did not overload the system, while optimized processing techniques contributed to faster analysis and response generation. This consistent performance highlights the system’s suitability for real-time web security analysis and practical cybersecurity applications.

C. Usability and Interface Evaluation

The graphical user interface was designed to be simple, interactive, and user-friendly, ensuring ease of use for both technical and non-technical users. Users were able to easily navigate between the Web Vulnerability Scanner and Phishing Defense Center modules through a well-structured layout and clearly defined sections. Features such as intuitive input fields, responsive control buttons, and real-time result displays significantly improved overall usability and user experience.

The dashboard presented organized and meaningful information, including requests, subdomains, open ports, detected vulnerabilities, and scan logs, enabling users to quickly understand the analysis results. Visual elements such as categorized outputs and structured result panels further enhanced clarity and readability. Additionally, the system minimized user effort by automating complex security checks, allowing users to perform detailed analysis without requiring advanced technical knowledge, thereby making the tool accessible for educational as well as practical cybersecurity applications.

D. System Reliability and Security Assessment

The system ensured reliable execution of phishing detection and vulnerability scanning processes. Error handling mechanisms were implemented to manage invalid URLs and network issues effectively. The scanning process was controlled through secure execution logic, preventing unintended interruptions. Generated HTML reports ensured consistent storage of results for future reference. The overall system maintained stability, accuracy, and reliability during repeated testing scenarios.

V. PERFORMANCE ENHANCEMENT

The performance of the proposed Phishing Website Detection and Web Vulnerability Scanner is improved through efficient Python-based processing modules and optimized scanning techniques. The graphical user interface ensures smooth interaction, while the backend handles real-time URL analysis, phishing detection, and vulnerability scanning with reduced processing time.

Real-time dashboard updates display scanning progress, detected vulnerabilities, and network details instantly, and features like Start, Pause, and Stop enable flexible task control. The system also supports automated HTML report generation for quick documentation. Overall, the integration of efficient processing, real-time analysis, and automated reporting enhances system performance and reduces analysis time compared to traditional methods.

VII. CONCLUSION

The proposed Phishing Website Detection and Web Vulnerability Scanner successfully demonstrates how cybersecurity techniques can be integrated into a single platform for efficient website security analysis. The system effectively identifies phishing websites and detects various web vulnerabilities through automated scanning and real-time analysis.

The developed tool eliminates the need for manual security checks and provides accurate results with minimal user effort. Real-time result visualization, combined with an interactive graphical user interface, enhances usability and enables users to monitor scanning processes efficiently. Features such as Start, Pause, and Stop controls, along with automated HTML report generation, improve flexibility and documentation.

The modular design ensures scalability and supports future enhancements, proving that the system can significantly improve website security assessment while maintaining accuracy, efficiency, and ease of use.

REFERENCES

[1] S. Garera , N. Provos, M. Chew, and A. D. Rubin, “A Framework for Detection of Phishing Websites”, Proceedings of the ACM Workshop on Recurring Malcode, 2007.

[2] M. Aburrous, M. A. Hossain, K. Dahal, F. Thabtah, "Intelligent Phishing Detection System for E-Banking Using Fuzzy Data Mining”, “Expert Systems with Applications, 2010.

[3] Y. Zhang, J.Hong, and L. Cranor, “CANTINA: A Content-Based Approach to Detect Phishing Websites", International World Wide Web Conference, 2007.

[4] R. Verma and K. Dyer, "On the Character of Phishing URLs Using Machine Learning”, Network Security Journal, 2015.

[5] A. Le, A. Markopoulou, and M. Faloutsos, “PhishDef: URL Names Say It All", IEEE INFOCOM Conference, 2011.

[6] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: A Fast Filter for the Detection of Phishing Websites", World Wide Web Conference, 2011.

[7] OWASP Foundation, "OWASP Top 10 Web Application Security Risks", 2021.

[8] W. Halfond, J. Viegas, and A. Orso, "A Classification of SQL Injection Attacks and Countermeasures", IEEE International Symposium on Secure Software Engineering, 2006.

[9] N. Antunes and M. Vieira, "Comparing Web Vulnerability Scanning Tools", International Conference on Dependable Systems and Networks, 2010.

[10] G. McGraw, "Software Security: Building Security In", Addison-Wesley Professional, 2006.

[11] Symantec Corporation, "Internet Security Threat Report", Symantec Security Response, 2019.

[12] Acunetix Ltd., "Web Vulnerability Scanning Techniques and Tools", Acunetix Web Security Report, 2020.

[13] IBM Security, "IBM X-Force Threat Intelligence Index", IBM Security Report, 2021.

[14] J. Mao, J. Bian, W. Tian, "Phishing Page Detection via Learning URL Features", International Journal of Network Security, 2018.

[15] S. Almomani, B. Gupta, A. Karimi, D. Al-Jarrah, "Phishing Website Detection Using Machine Learning Techniques", International Journal of Information Security, 2013.